

PI PICO 2 NEW!
We go hands-on &
hack Wi-Fi on to it

FAST GRAPHICS
Boost your 3D speeds
with an Nvidia GPU



ENIGMA TODAY
Revive using the WW2
encryption system

LINUX

FORMAT

The **#1** open source mag



START USING LINUX

We test the best beginner
distros on the planet!

BRIGHTER SMARTER HOMES!

Build a brilliant open home
lighting system that you can
design, grow and upgrade

100
pages of Linux
tricks, tips
& more!

PLUS: HOW TO

- » Run your own C64 assembly code
- » Emulate the classic Psion 3 computer
- » Fix & tweak your Ubuntu network settings

BREADBOARD OS

Get using the real-time
OS for all Pi Pico builds

BUILD A SHELL

We're coding our very
own Linux terminal

INSIDE THE AI PC

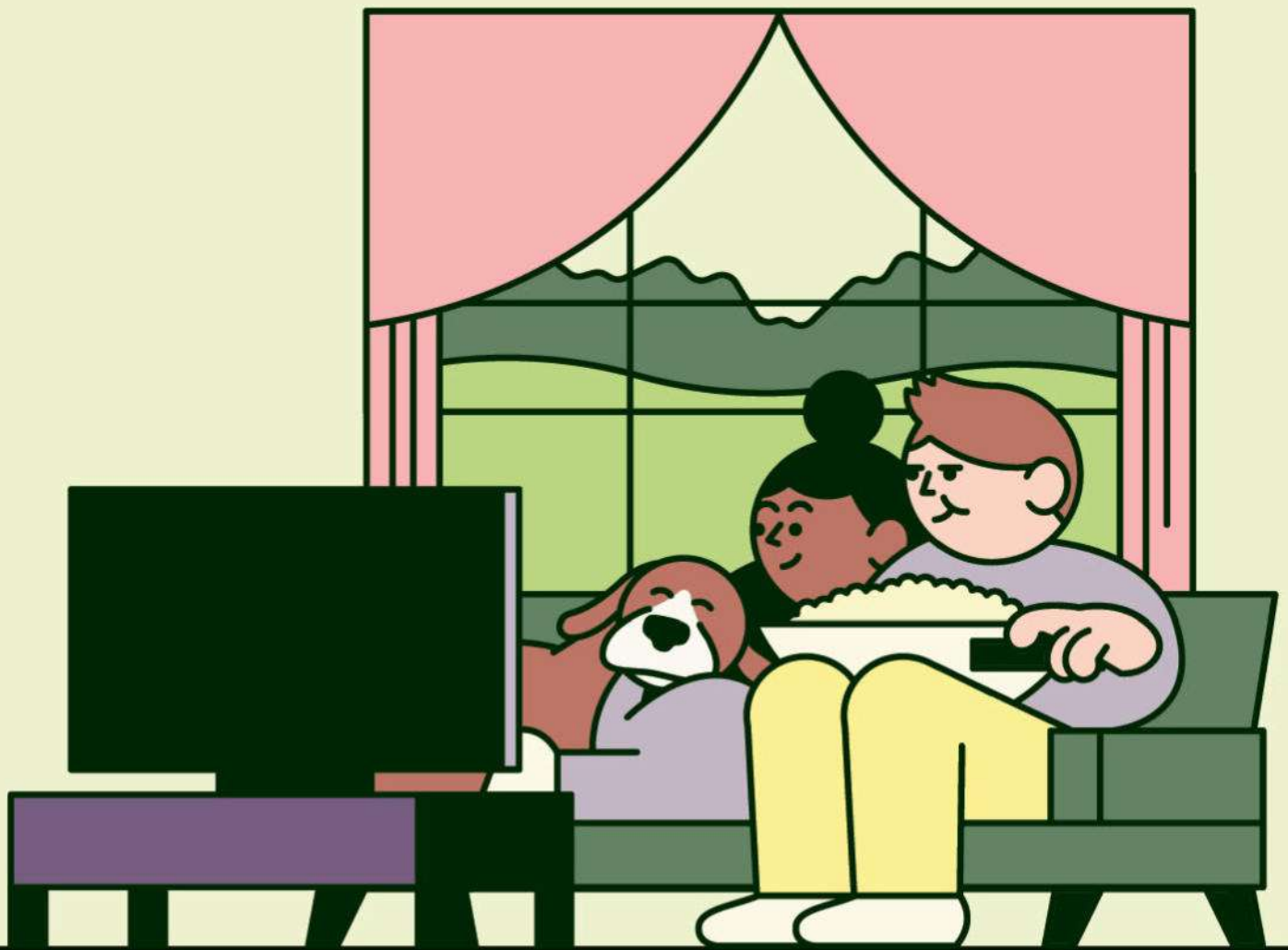
What the heck is the
new AI PC standard?

LXF October 2024



SAVVY SAVERS, STEP THIS WAY.

Discover cover options for car, home and travel insurance, plus broadband and more.



GO.
COMPARE

Get more information and compare quotes at [Go.Compare](https://www.go.compare).



LINUX FORMAT



» MEET THE TEAM

This issue, we are lighting up our lives with Home Assistant-powered smart lights, so what's the smartest thing you have installed lately?



Matthew Holder

The smartest thing in my house, is certainly not me! The Raspberry Pi 4 that runs my *Home Assistant* installation and camera management has been a true workhorse and deserves plenty of praise. This hub

enables everything else to happen, automating all the things.



Michael Reed

My house is mostly analogue – I press the light switch myself. As for 'smart' tech, an app (*Sleep As Android*) monitors my sleep. Heart rate and steps are recorded with a fitness band (Amazfit 5), while *Fitotrack*

records runs. I visualise the data with Python/Matplotlib.



Nate Drake

I recently encountered a smart toaster that pings an app when both rounds are a perfect shade of golden brown. Now I'm not just living the dream – I'm toasting it. These little slices of heaven are always Insta-ready. My only regret is it isn't a

'Talkie' like the one in *Red Dwarf*.



Les Pounder

The smartest thing that I have recently made is a Raspberry Pi Zero 2 W-powered Wi-Fi-enabled microscope because I needed one for a Tom's Hardware story on the RP2350 found in the new Pico 2.

So, I used a Pi to photograph a Pi for a Pi story.



Nick Peers

I set up my own smart home surveillance system – not by buying into any proprietary tech, but through the brilliant open source *Motion*, combining a ragtag of webcams, IP cameras and old phones for comprehensive coverage of what our cats get up to while we're away.

*Savings are based on the cover price.

Not so smart homes



Every tech startup and their dog seems to want to start a smart-home ecosystem, which is terrible because you end up with a smorgasbord of apps trying to independently control each aspect of your home. Amazon, Google and Apple have swooped in and tried to corner their own patch of your smart home, which helped streamline control but you're at the whim of Big Tech and its monitoring.

Open protocols like Zigbee have helped consolidate control: buy your smart bulbs, install smart switches, wire inline controls to the mains and centrally control them, but how? We're taking the mighty open source *Home Assistant*, a Pi and a Zigbee dongle and will explain just that, so you can have a glowing fully lit smart home in time for the dark winter nights.

We've also got our first hands-on with the brand new Pi Pico 2. With its custom-designed silicon, it's going to be powering a new generation of home projects. Turn to page 45 to get Les Pounder's opinion on the early builds and a project to give it wireless abilities. If this sounds a little too much like hard work, we've also tested the best beginner distros, hark back to the Psion 3 and 5 days, relive using the Enigma machine, and take an early look at the new wave of AI PCs that everyone will be trying to sell you with that new CoPilot key to enjoy!

Neil

Neil Mohr Editor
neil.mohr@futurenet.com



Subscribe
& save!

On digital and print:
see page 16

Contents



REVIEWS

File Edit View Actions Business Reports Tools Windows Help		
Accounts		
Accounts Name	Description	Total
Assets	Assets	£1,041.34
Current Assets	Current Assets	£1,041.34
Cash in Wallet	Cash in Wallet	657.34
Checking Account	Checking Account	£974.00
Savings Account	Savings Account	10.00
Equity	Equity	£1,041.34
Opening Balances	Opening Balances	£1,041.34

GnuCash 19

The words “free accounting” are music to **Rob Clymo**’s freelance ears, but is this money management app feature-rich?

Vanilla OS 2 Orchid 20

Is Vanilla OS a darling bud of May or just an awkward orchid? **Nate Drake** is on the case.



Emmabuntüs DE 5 1.02 21

Nate Drake searches frantically for his umlaut key in evaluating this ethical, lightweight and accessible French distro.

Sleeper OS Bee v2 22

Nate Drake fires up his laptop to discover whether Sleeper OS is the perfect dream or every Linux user’s nightmare.

Super Grub2 Disk 2.06s4 23

Is it a bird? Is it a plane? No, it’s Super Grub! **Nate Drake** shares how it worms its way into unbootable systems.



Schim 24

The Management is thrilled to hear that **Jody Macgregor** is happy working in the shadows.

BRIGHTER SMARTER HOMES

Matt Holder lights up our lives with Home Assistant and Zigbee devices to create automations, useful dashboards and easy control over our homes. See page 32!



ROUNDUP



Beginners’ distributions 26

Not everyone is a Linux veteran like **Michael Reed**, so he’s assessing five distributions that could be a good choice for Linux newbs.

IN DEPTH



Inside the AI PC 78

Microsoft, AMD and Intel are all pushing the idea of AI PCs, but are they anything more than marketing speak? **Barry Collins** and **Tim Danton** find out...

PI USER



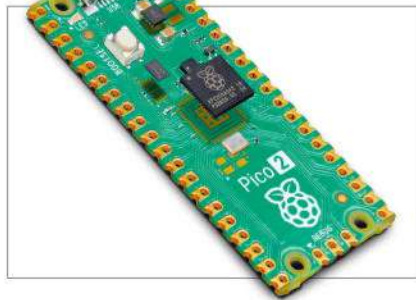
Pi news 43

Les Pounder is impressed with the latest Intel-based Raspberry Pi rival, plus news stories from the single-board world.



Elegoo Saturn 4 Ultra 44

It turns out that **Denise Bertacchi** will go crazy for anything that tilts! Like this impressive 3D printer.



Pi Pico 2 45

With a £5 note burning a hole in his pocket, **Les Pounder** splashes out on the new embedded Pi.

Hack Wi-Fi on to your Pi Pico 2 46

The new Pico 2 doesn't come with Wi-Fi, but when has that ever stopped **Les Pounder**? Never, that's when.

Cut a slice of Pi on BreadboardOS 48

Filling up on tasty carbs is one way **Tam Hanna** spends his weekends, when not making questionable devices in his lab.

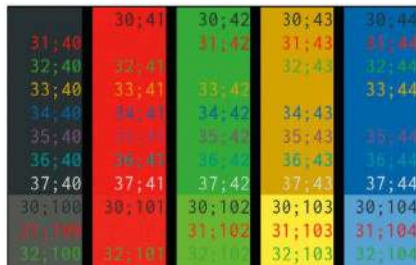
CODING ACADEMY

Add colour to the LXF Shell 90

Removing his head from the retro demoscene clouds, **Ferenc Deák** is back down to earth to add colour to our project.

Create a VS Code 6502 extension 94

David Bolton shows how to set up *Node*, *NPM* and *Yeoman* for creating extensions for *Visual Studio Code*.



REGULARS AT A GLANCE

News 6

Canonical climbs high; new *GIMP* creeps closer; zeroing in on 0.0.0.0 Day vulnerability; Rust adoption tarnished; Linux outshines Windows; plus industry opinion and a look at new distro releases.

Kernel watch 10

Answers 11

Neil Bothwick regularly backs up his brain to an external drive, which is how he can readily answer questions about automounting drives, overcoming public Wi-Fi limitations, and much more.

Mailserver 14

Editor **Neil Mohr** shares and receives advice from *Linux Format* readers.

Subscriptions 16

Grab your monthly dose of Linux and save a massive 50% off the usual price!

Back issues 60

Get hold of previous *Linux Format* editions.

Overseas subscriptions 61

Get *Linux Format* shipped around the globe.

HotPicks 83

Mayank Sharma isn't any good with a fishing line, but he knows how to trap and catch the latest and greatest free and open source software, such as *VirtualBox*, *Signal*, *GpgFrontend*, *Alpaca*, *Keypunch*, *Falkon*, *A-Mazing Urho*, *Etcher* and more.

Next month 98

TUTORIALS

TERMINAL: Speed up downloads 52

Optimum use of internet bandwidth is **Shashank Sharma's** life's purpose. Thankfully, there are tools that help.

LINUX BASICS: Network settings 54

Nick Peers uncovers the mysteries of how to set up, troubleshoot and improve network performance.

ONLYOFFICE: Advanced PDF tricks 58

PDF is a hugely popular format, touching our workflow every day. **Kseniya Fedoruk** shares tips to get the most out of PDFs.



EMULATION: Classic Psion PDA 62

Les Pounder travels back to the 1990s, when Filofax became digital, and your phone was dumb and the size of a brick.

ENCRYPTION: WW2 Enigma machine 66

The Enigma machine was sophisticated for its day, but its mechanics can be replicated in software now, as **Mike Bedford** reveals.

NVIDIA: Upgrade your graphics card 70

Despite the company being famously anti-open source, **Neil Mohr** explores how to easily upgrade to an Nvidia graphics card.



ADMINISTERIA

UNIT FILE	STATE
proc-sys-fs-binfmt_misc.automount	static
+.mount	generated
boot.mount	generated
dev-hugepages.mount	static
dev-nqueue.mount	static
proc-sys-fs-binfmt_misc.mount	static
sys-fs-fuse-connections.mount	static
sys-kernel-config.mount	static
sys-kernel-debug.mount	static
tmp.mount	disabled
systemd-ask-password-console.path	static
systemd-ask-password-plymouth.path	static
systemd-ask-password-wall.path	static
session-1.scope	transient

Administeria 76

Stuart Burns has top tips for sysadmins to help fix broken services on Linux, and applauds the new Pi Pico 2.

Newsdesk

THIS ISSUE: Canonical climbs high » New GIMP creeps closer
» Zeroing in on 0.0.0.0 Day » Rust adoption tarnished

DISTROS

Canonical is having a great time with Ubuntu

The company has recently filed its financial statements. It saw \$251 million in revenue for the last year, and has 1,034 employees.

As a limited liability company, Canonical annually submits financial reports to Companies House in the UK. The most recent reveals a total turnover of \$251 million in 2023, a \$46 million improvement on the previous year. The overall profit for the financial year was \$12.5 million, a considerable increase from \$3.9 million in 2022.

The figures are particularly impressive given that just five years ago Canonical reported that it was operating at a loss, with only around half the revenue and employees it enjoys today. Speaking of personnel, from 2022 to 2023 the staff also increased from 858 to 1,034.

Part of this success is no doubt thanks to businesses paying for Ubuntu Pro plans, although subscriptions remain free for personal use. Canonical also apparently has other services designed for enterprise-level Ubuntu up its sleeve, such as the recently released Ubuntu Core.

In mid-August, Canonical also made the very welcome announcement that as of Ubuntu 23.10, the operating system will use more up-to-date Linux kernels.

Traditionally, Ubuntu has followed a hard time-based release process, with release dates committed six months or more in advance. This has always been at odds with the process followed by the upstream kernel developers, who generally produce a stable release every two to three months.

The CKT (Canonical Kernel Team) has also usually required around one month between an

upstream release and the associated Ubuntu kernel to be considered stable enough for release. This caused huge issues when the kernel was released a few weeks before or after a stable version of Ubuntu.

This is exactly what is anticipated to happen, given the upstream release of version 6.11 of the Linux kernel is currently scheduled to be available around the same time as Ubuntu 24.10.

Canonical has decided that in future it will ship the latest upstream Linux kernel available at the time of the Ubuntu release freeze date, even if it's still in RC (Release Candidate) status.

This is good news for users who want to take advantage of the latest features. Ubuntu will also offer a 'bridge kernel' for users who wish to upgrade to a pending release but require dependent components that have not yet stabilised. This bridge kernel ensures that users can maintain system performance while waiting for the fully stable kernel to be released.

Speculation remains rife as to whether Canonical will go all out and launch an IPO, in the vein of Red Hat.



Mark Shuttleworth has been driving Canonical to be profitable with changes instigated in 2017.

SHIPPING FORECAST

In future it will ship the latest upstream Linux kernel available at the Ubuntu release freeze date, even if still in RC status

SOFTWARE

GIMP 3.0 is coming



The much-anticipated GIMP 3.0 will soon be upon us, with the latest developer preview build announced.

On 20th August, *GIMP* entered 'string freeze' for version 3.0.0. The following day, the development preview (2.99.18) was released for testing purposes. The stable version of 3.0 is likely still a few months away but the development release has tantalised us with some major and very welcome changes.

Chief among these is a revamped welcome screen, no doubt in response to criticism that *GIMP* can be very unfriendly to beginners. It has been overhauled with four new sections. These include a Personalise section to easily change features such as colour and icon themes. The Create section displays the eight most recently accessed images, with the option to launch any of them. There's also a Contribution section and a link to easily access the release notes.

GIMP will also now include non-destructive layer effects. If you're unfamiliar with graphic design, this is the ability to change output pixels

GIMP 3.0 includes an option for auto-expanding layers, which adjust their size instantly if you paint outside the boundaries.

while keeping the source pixels intact. From now on, *GIMP*'s filter effects, such as Blur, will be kept separate from a layer's pixels. Any GEGL (Generic Graphical Library) option that has a GUI will also apply to layers non-destructively.

GIMP also now has a new auto-expanding layers feature. All Brush tools will have an extra option to Expand Layers. When checked, if you paint past the layer boundaries, they will automatically expand without you having to adjust the layer size itself.

GIMP has new snapping options, too, for aligning layers on the canvas. You only need to enable Snap To Bounding Boxes to display dynamic guides so you can check that the layer you're moving is aligned with the centre or sides of others.

SECURITY

The 0.0.0.0 Day vulnerability

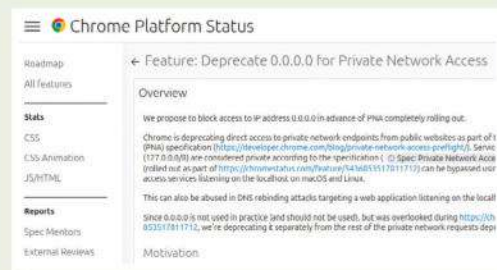
The exploit allows execution of arbitrary code.

In early August, Oligo Security researchers publicly disclosed the 0.0.0.0 Day vulnerability. This exploit, which impacts Mac OS and Linux devices, allows websites crafted by bad actors to bypass browser security and interact with local network services. The implications are serious, given that attackers could potentially remotely execute any arbitrary code.

Both *Chrome* and *Firefox* allow external websites to access local resources. However, Windows versions of the browsers aren't affected as the OS itself blocks the IP 0.0.0.0.

Google's PNA (Private Network Access) feature is designed to prevent exploits of this kind, but while PNA blocks 127.0.0.1 and certain other localhost addresses, the 0.0.0.0 exploit can bypass its security features.

Technically, any application that runs on localhost and can be reached by 0.0.0.0 is



Google has plans to deprecate 0.0.0.0 in PNA but it will be harder to implement in *Firefox*.

vulnerable to remote code execution. Oligo researchers found that the recent Shadowray attack could be carried out using this exploit.

As of *Chrome* 128, access to 0.0.0.0 has now been blocked. Mozilla has changed the fetch specification for *Firefox* to do the same, but as it doesn't implement PNA, a permanent fix may still be some way away.

OPINION

LATEST UPDATES!



Italo Vignoli is one of the founders of LibreOffice and the Document Foundation.

“ LibreOffice 24.8 has just been announced, with several last-minute new features that were not yet ready when we covered the major release in August.

This is the first major release available for ARM-based Windows PCs in addition to Intel and AMD-based PCs, so *LibreOffice* is now available for all flavours of Windows PCs.

In terms of privacy, if the option Tools > Options > LibreOffice > Security > Options > Remove Personal Information On Saving is enabled, any personal data is not exported.

Calc adds the following new functions: FILTER, LET, RANDARRAY, SEQUENCE, SORT, SORTBY, UNIQUE, XLOOKUP and XMATCH, and improves performance for threaded calculations and optimisation of redraw after a cell change by minimising the area to be refreshed.

Impress enables you to scroll between slides in Normal view, while the notes are available as a collapsible pane under the slide. Also, the running slideshow is now immediately updated when applying changes in Edit view or Presenter Console, even on different screens.

End users looking for support will be helped by the immediate availability of the LibreOffice 24.8 Getting Started Guide: <https://books.libreoffice.org>. **”**

OPINION

TURNING
THE TDE

Dave Stokes
is a technology evangelist
at Percona.

“You might not get excited about encryption, but it matters. Organisations around the world hold data on individuals and companies, and it needs to be kept secure in motion and at rest.

However, we don't have all the tools we might like to work with. *PostgreSQL* lacks transparent data encryption (TDE), which allows users to work with data transparently and easily without sacrificing security. Proprietary counterparts like *Oracle* and *SQL Server* have this feature, yet *PostgreSQL* encryption is limited to cluster-wide implementations.

Implementing TDE in *PostgreSQL* is difficult, as it involves a change in the core project that could affect or break other functionality.

At Percona, we are developing a new open source *Postgres* extension – *pg_tde* – to encrypt table contents. This is not full TDE, but *pg_tde* will provide an example of what can be achieved with the current *Postgres* extension framework, and where we can extend extensions to provide more functionality where users need and want to use it.

If the proposed changes prove successful, per-table TDE may soon be available in the world's most advanced and sought-after open source relational database. ”

KERNEL

Rust adoption lags
behind expectations

Linus Torvalds attended KubeCon in Hong Kong to speak about AI and Rust.



Torvalds came mainly to speak about AI adoption in Linux but also expressed his concerns with Rust implementation in the kernel.

Linus Torvalds recently expressed frustration over the slow adoption of Rust in the Linux kernel at KubeCon.

Speaking to a packed crowd, he emphasised he was there for the Open Source Summit. He also acknowledged: “The very slowly increased footprint of Rust has been a bit frustrating,” citing reluctance among “old-time” kernel developers who are more comfortable programming in C.

He also pointed out that the Rust infrastructure itself has struggled with stability

issues, further hindering its integration. “In the last release, we finally got to the point where the Rust compiler we can use for the kernel is the standard upstream Rust compiler,” he said. This may signal a move towards Rust stability, but he noted, “We're not there yet.”

Despite the challenges, Torvalds remains hopeful, emphasising that the initial hurdles are being overcome, albeit slowly. “I'm hoping that we're over some of the initial problems,” he added, but acknowledged that it has taken longer than anticipated.

CREDIT: Wikimedia

GRAPHICS

Nvidia fixes
Wayland

The latest driver supports KDE and direct scanout on multiple monitors.

Nvidia's latest Linux driver, v560.35.03, released on 21st August, addresses some persistent issues with Wayland, enhancing system stability and performance.

Notable fixes include resolving KDE Plasma Shell freezes when running in Wayland compositor mode or while hovering over or opening applets.

The latest driver eliminates display freezes during Wayland direct scanout on multiple monitors. It also fixes kernel crashes related to DRM operations and corrects a bug that would often cause crashes in Xwayland games.

Read more at www.nvidia.com/en-us/drivers/details/230918/.

The updated Nvidia driver enhances system support for Wayland in supported devices.



OPERATING SYSTEMS

Linux
outperforms
Windows (again)

Clear Linux outperforms Windows on AMD Ryzen 9 9950X hardware.

Linux continues to outperform Windows, especially on AMD Ryzen 9 9950X hardware, as demonstrated in benchmarks by Linux engineer and *Phoronix Test Suite* developer Michael Larabel.

His latest testing focused on using the AMD chip with Arch Linux, CachyOS, Clear Linux, Fedora Workstation 40, Ubuntu 24.04 LTS and a snapshot of Ubuntu 24.10.

Larabel's tests found that Clear Linux performed best out of the box, due to features like compiler function multi-versioning and performance-minded defaults.

Full results are available at www.phoronix.com/review/linux-os-amd-ryzen9-9950x.



The Ryzen 9 9950X is AMD's current flagship CPU, with 16 cores, 32 threads and a 4.3GHz base clock.

CREDIT: Nvidia, AMD

Distro watch

What's behind the free software sofa?

EXTIX 24.8

ExTiX is an older Linux distro, first released in 2008. The OS is based on the Chinese distro Deepin. It can be run as a live CD by booting the OS entirely into RAM. The latest version doesn't include *VirtualBox* Guest Additions but the developers claim you can resize the screen anyway from within the OS. The *Refracta* installer has now been replaced with Deepin's own, which offers better support for installing GRUB on UEFI machines. Read more at www.extix.se.



ExTiX has better VM support, a new browser and installer.

REBECCABLACKOS 2024-08-12

This OS's anonymous developer confessed he "got the idea to name a Linux distribution after my favourite celebrity after I saw some earlier Linux distributions named after celebrities". Unlike the Justin Bieber-inspired Biebian and Hannah Montana Linux, this distro is functional and based on Debian 12 (Bookworm). Its primary purpose is to be a live distro that provides a Wayland server. The latest version has removed X from the system startup path. Mercifully, 'Friday' is not preinstalled. See <https://sourceforge.net/projects/rebeccablackos/>.



This celebrity-themed distro can run Wayland desktop sessions.

IPFIRE 2.29 CORE 187

Since 2011, IPFire has provided a solid firewall OS with an emphasis on easy setup and maintenance. The latest version of IPFire can now use TCP SYN cookies to stymie SYN flood denial-of-service attacks. It also creates more rules in the most restrictive mode to allow IPsec traffic for local connections. As Intel's Hyperscan library is no longer free software, it's now been replaced with the open source fork Vectorscan. You can learn more at www.ipfire.org.



IPFire enhances security with SYN cookies.

TAILS 6.6

This Debian-based OS's stated goal is to provide complete internet anonymity for users. All internet connections are routed through the dark net. The developers also encourage users to use Tails in live mode. Nevertheless, the operating system has a persistent storage feature, which has been upgraded in the latest version, so it's now simpler to create persistent storage on first boot. Storage settings also now don't freeze when users open links to documentation. Find out more by visiting <https://tails.net>.



Debian-based Tails OS boosts user anonymity.

CACHYOS 240714

This distro is based on Arch with optimisations for speed and security. It also has a heavily modified version of the kernel, care of the *BORE* (*Burst-Oriented Response Enhancer*) scheduler. The latest CachyOS introduces open Nvidia modules by default. It also offers the Cosmic desktop and improves connection speeds via a worldwide cache CDN. Other upgrades include a Secure Boot script, as well as automatic mounting in *cachy-chroot* when using the live ISO. <https://cachyos.org>



The latest release improves hardware detection and more.

OPINION

TAMING PANTHOR



Eric Smith is a senior software developer at Collabora.

“The Panthor kernel driver and Mesa Panfrost driver combination recently passed an important milestone. It is now officially conformant for OpenGL ES 3.1 on Mali-G610 (a chip in the 10th generation of Mali).

This required passing tens of thousands of test cases. Passing these gives hardware implementers and software adopters assurance that the driver is sound. This doesn't mean that Panthor and Panfrost are bug-free, of course. Software being software, there are always bugs. It does mean that it's unlikely that there are still major issues. Users can be confident that programs using OpenGL ES 3.1 will 'just work' on the tested version of Mesa (24.11).

This milestone is naturally just another marker on a long journey. Panfrost had already achieved conformance on Mali G52, as described by Alyssa Rosenzweig. Various other Mesa projects have passed conformance tests, too (such as Zink and NVK).

Looking to the future, we are working on improving Panfrost and Panthor support for all generations of Mali devices. And while the OpenGL ES implementation has passed conformance tests, there is still room for new features and improvements. The future is looking bright for Linux graphics on Arm-based GPUs!

OPINION

BOOT
CAMP

Jon Masters is a kernel hacker who's been involved with Linux for over 22 years, and works on energy-efficient Arm servers.

“A recent security update from Microsoft intended for Windows-only installs had the (unfortunate) side effect of bricking some Linux dual boot installs. Microsoft was shipping a patch that told UEFI boot firmware not to trust certain known insecure bootloaders that might break Secure Boot.

Secure Boot on Linux relies on signed bootloader code that has been blessed by Microsoft. In a typical system, a small shim binary is loaded prior to the Linux bootloader GRUB. The shim is signed by Microsoft and has what is called security generation. When an incident arises, the security generation is bumped and Microsoft can issue an update telling firmware not to trust software with an earlier generation.

Some distros had quite out-of-date builds of these early boot components and failed to boot once the Microsoft updates had been deployed. Microsoft meant to skip updates on dual boot systems, but something didn't work as expected. For once, I don't think we can fault Microsoft. It was trying to make machines more secure by distrusting older known insecure software. It's just unfortunate that this turned out to include a number of popular distros.

Kernel Watch

Jon Masters summarises the latest happenings in the Linux kernel, so that you don't have to.

Linus Torvalds announced Release Candidate 5 (RC5) of what will become Linux 6.11 very soon now.

At this rate, we will be covering the merge window (period of time during which disruptive new features are merged into the kernel) for 6.12 in next month's issue. That being said, there aren't looking to be as

“nontechnical nonsense”, which might be more bluntly described as obstinate pushback from some of the kernel developers who may see Rust to be an annoying distraction that must be supported in parallel with existing C language support.

The Rust developers would, of course, promptly respond (and they did) that there is no

attempt to force anyone to adopt Rust for future kernel development. They instead argue that they know they have a slog ahead and intend to add support piecemeal in terms of ABIs and then consumers (drivers and so on) of those ABIs. Nonetheless, the latest development sparked a great deal of discussion in some of the usual forums,

CORROSIVE TALK

“There are certainly always opinions on the internet, but this is one of those cases where opinions are very strong and feelings can be raw.”

many big ticket items landing in 6.12 as there were in the previous release.

Rust for Linux

One topic under very heavy discussion as we go to press is that of one of the lead developers behind the Rust for Linux project choosing to ‘retire’. Wedson Almeida Filho announced their departure via a short patch cover letter (an email describing a patch series) that simply removed them from the maintainers. The stated reason for leaving is what has garnered attention and discussion:

including a Linux Weekly News (LWN) post with nearly 200 replies so far. There are certainly always opinions on the internet, but this is one of those cases where opinions are very strong and feelings can be raw.

Speaking of Rust, assorted other ongoing developments include the rewriting of modversions (the code that generates symbol versioning information for kernel modules, used by *modprobe*, *insmod* and other *kmod* tools to determine whether a module is compatible with the running kernel) to support Rust-based modules. **LXF**

» ONGOING DEVELOPMENT

Lizhi Hou from AMD posted an AMD XDNA driver. This includes support for the NPU (neural processing unit) aka AI inference accelerator included in upcoming AMD client CPUs. This is the kind of accelerator found in AI PCs of the form announced by Microsoft with much fanfare (see p.78). On Linux, the NPU driver allows the various AI/ML frameworks (PyTorch, TensorFlow and so on) to be wired up to the NPU.

James Gowans from Amazon posted a new filesystem driver called *guestmemfs* used to preserve the state of running guest

virtual machines when using *kexec* on the host system to upgrade the kernel while the VMs are running. This might be of interest to enterprise distro folks in addition to the cloud use cases.

Mark Brown posted an updated patch series that allows applications to control the setup of shadow stacks. These are used to support a hardware feature on various CPUs that shadows (copies) parts of the main app stack on function call entry/exit in a manner that allows for detection of stack smashing and overflow attacks.

Answers



Neil Bothwick
regularly
backs up his
brain to an
external drive.

Got a burning question about open source or the kernel?
Whatever your level, email it to answers@linuxformat.com

Q Filling fstab

I'm on Ubuntu 22.04 and using it for a lot of video/audio projects. That's resulted in the accumulation of a lot of drives. Since I often have dormant projects mixed with active as well as archived ones, my drives are not always plugged in. Until now, I've just used my *Dolphin* file manager to mount a drive by simply clicking it. With a bunch of drives, though, it can get to be a nuisance.

I'd like to have the system, when booting, auto-mount whatever drives are plugged in (and ignore any not plugged in). I'm assuming this might be accomplished by listing the drives in **fstab** and that it may be time to start entering all of them. Is **fstab** a good way to handle this? The listing of drives in **fstab** could get quite long. Is that bad?

Jay Robson

A It can be done with **fstab**, but as you mention *Dolphin*, it sounds like you use a KDE Plasma desktop and that provides a simpler solution. Open System Settings and go to Hardware > Removable Storage > Removable Devices. There you will see an entry for All Known Devices. Tick the On Login or On Attach box (or both) and any drives you have previously

used will be automatically mounted when they are plugged in, while any new devices still ask whether you want to mount them.

It sounds like this is what you need, but if you want to use **fstab**, or you are using a different desktop, that is a way to do it. It could make for a long **fstab** file if you have a lot of drives, but that's not a problem as long as you keep it organised. Keeping all your removable drives together, along with sensible use of comments and blank lines, will keep the **fstab** file readable. As the devices could be plugged in at any time, their device nodes could change, so you need to identify them by UUID or LABEL. If possible, give each of your filesystems a unique and meaningful label, to keep **fstab** readable. Otherwise you need to use UUIDs and comment each entry.

You also need to change the options for each drive and there are two approaches you could take. One way is to add **nofail** to the options for each drive. This means the system will try to mount all the drives when you boot but will not complain about missing ones. If you subsequently connect a drive, you will need to issue the mount command manually, so best add **user** to the options to avoid needing root.

It works but feels a little kludgy. The other option is to use automounting, by

adding **x-systemd.automount** to the options for the drive. In this case, the drives are not mounted when the system boots or when they are plugged in, but when you access the mount point. So, for a drive with the rather unimaginative filesystem label of **Data**, you could use either of the following:

```
LABEL=Data /mnt/data auto
nofail,user,other,options 0 0
LABEL=Data /mnt/data auto x-systemd.
automount,other,options 0 0
```

In either case, and in contrast to the way in which desktop automounters work, the mount points need to be created in advance. Also bear in mind that you will have to take care of unmounting the drives before removal.

Q Ports of call

As something of a road warrior, I often use the Wi-Fi in the likes of coffee shops. The trouble is that they often block many ports, meaning I can't SSH into servers I need to work on or join VPNs using OpenVPN. As they have to allow ports 80 and 443 for people to be able to use the web, is there a way to access other services through these ports?

Oliver Lawson

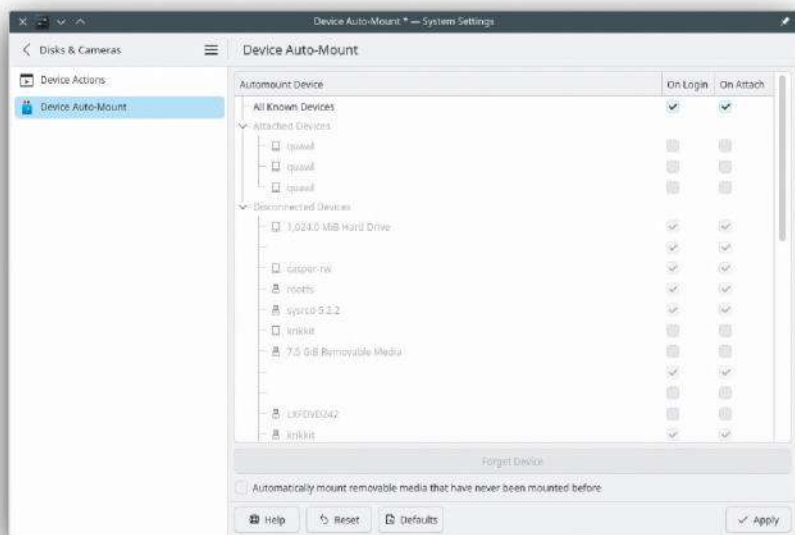
A You can do what you want, but how you do it depends on your exact needs and whether you are already using these ports to run a web server on the remote system. If the ports are free, you can simply set up SSH to allow connections on a different port, using the **Port** directive in **/etc/ssh/sshd_config**. You can listen on more than one port, so the config file could include:

Port 22

Port 443

That way, you can still use port 22 as normal and use 443 when you have to. Obviously, you will have to forward port 443 on the router.

If port 443 is in use on the remote computer, such as if it is running a web server, you can use **ssllh** (www.rutschle.net/tech/sslh/README.html) to handle this. This software listens on a port and



To save overloading your **fstab**, KDE Plasma can automatically mount only those drives it knows about.

redirects connections to the correct server depending on the contents of the first packet. The standard setup can differentiate between HTTPS, SSH and OpenVPN connections and forward them accordingly.

Once installed, the program can read configurations from `/etc/ssh.cfg`, but for a simple setup, you can do everything on the command line. For example:

```
$ ssh -p 192.168.1.8:443 --ssh 127.0.0.1:22
--http 127.0.0.1:80 --ssl 127.0.0.1:443
--openvpn 127.0.0.1:1194
```

The `-p` option specifies the interface and port to listen on. For a server connected to the internet via a NAT router, this is the LAN address of the machine. This example then forwards SSH, web (plain and SSL) and VPN traffic to the relevant port on the localhost interface. If you get an error about a port already in use, you need to alter the server configuration to listen on localhost only. For Apache, this means your configuration files in `/etc/apache` would contain:

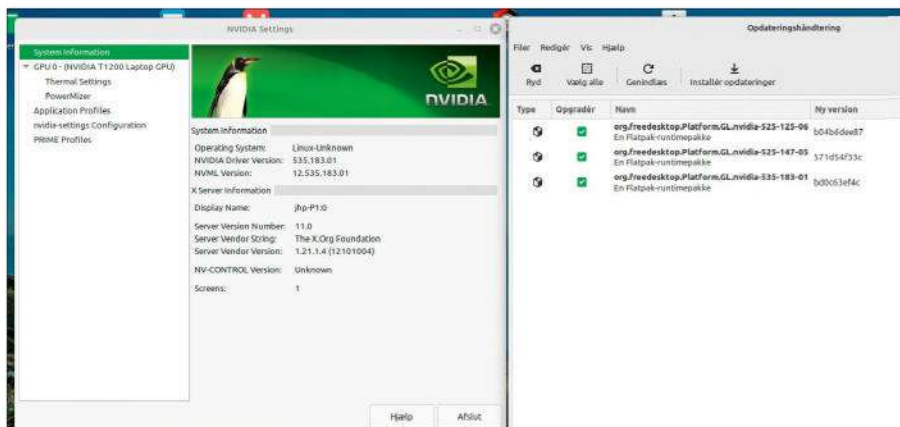
```
Listen 127.0.0.1:443
Listen 127.0.0.1:80
```

These examples all forward to localhost, but you can forward to any computer that is accessible from the server running `ssh`.

Q Nvidia bloat
What is `org.freedesktop.Platform.GL.nvidia`? I am running Mint 21.3 on a Lenovo Thinkpad P1. In the Mint updates, they keep on emerging in the update list, apparently taking up a lot of space and reappear even after an 'update all' run.

Searching the internet, I have not been able to figure out exactly what they do or whether they are needed or not. It seems that they also address Nvidia versions I do not have.

Jorn Pedersen



Too many Nvidia Flatpak packages appears to be a common problem, but you can remove them manually.

A These are Flatpak packages that provide versions of the Nvidia drivers and libraries. They are there because Flatpak packages do not access system libraries; they only use libraries in their own or another Flatpak. You almost certainly have some software installed via Flatpak that uses OpenGL, so an Nvidia Flatpak is needed for it to gain access to the Nvidia GL libraries.

Normally, you would get rid of Flatpaks that are no longer needed with:

```
$ sudo flatpak remove --unused
```

In this case, it will tell you that those Flatpaks are pinned and will not be removed. It should be safe to remove all but the version corresponding to your Nvidia driver with `flatpak uninstall`. First list the installed versions with:

```
$ sudo flatpak list | grep nvidia
```

Then you can remove all but the current one with:

```
$ sudo flatpak uninstall org.freedesktop.Platform.GL.nvidia-525-125-06
```

You can either remove them one at a time or specify them all on the same command line. If you experience problems

after doing this, you can run:

```
$ sudo flatpak repair
```

However, the usual advice about having backups of everything import before making system changes applies.

Q Flashy names
I am learning to use Cinnamon 21.3. An 8GB flash drive is on the computer and shown on desktop but the icon says '8GB Volume'. I want to rename it to something indicating the contents – how do I do this?
William Bates

A Strictly speaking, USB flash drives, or any storage device, do not have names. The filesystems on them do. With a flash drive where the whole device is a single filesystem, usually contained in a partition that fills the device, it appears the same, but there is a difference. This means you need to rename the filesystem. Filesystems contain an optional label that is set when it is formatted or created. This is what the desktop shows – you are seeing '8GB Volume' because there is no label on the filesystem.

» A QUICK REFERENCE TO... SSH TUNNELLING

An SSH tunnel is an SSH connection between two computers that is used to transfer data securely between the two. For example, the following command results in all connections to port 8080 on the local computer being forwarded to port 80 on **example.com**:

```
$ ssh -L 8080:localhost:80 example.com
```

There are various reasons you may want to do this. There may be a restriction

that prevents you directly connecting to port 80 on the remote computer with a browser, for instance. Or you may wish to send sensitive data with a program that does not provide its own encryption. All traffic going through the tunnel is encrypted by SSH. A good example is the VNC problem answered above, VNC is not encrypted but you may not want anyone to be able to snoop on what you are doing on your desktop.

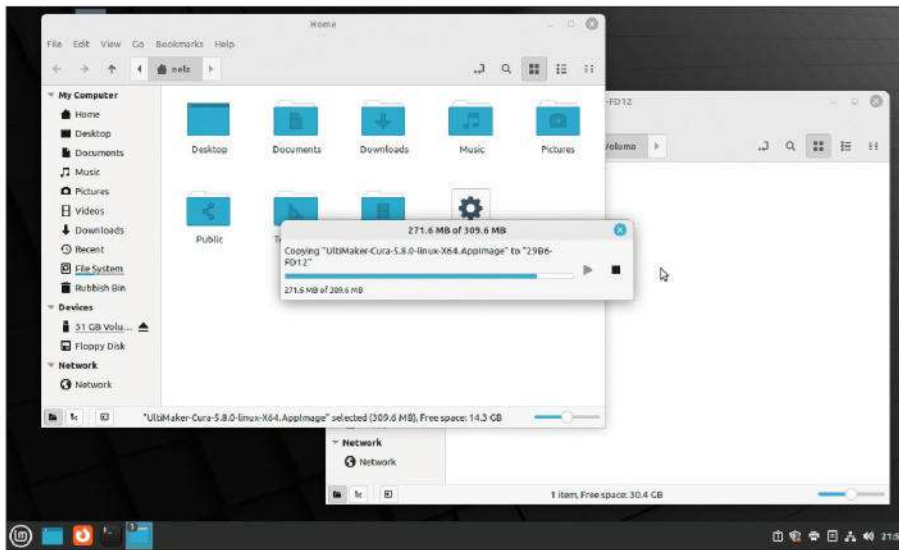
The above example uses the `-L` option to set up local port forwarding, from a port on the local computer to one on the remote system. You can also set up a reverse tunnel to facilitate connection in the opposite direction, with the `-R` option, as follows:

```
$ ssh -R 8080:localhost:80 example.com
```

In this case, we are connecting port 8080 on the remote system to port 80 locally.

Although we have set up a tunnel through localhost, the tunnel allows connections from any computer. Another computer on your LAN can connect to port 8080 on your computer in the first example to access the remote system. You can prevent that by adding a bind address – so, to only allow the tunnel to be used by the system running it, you could do the following:

```
$ ssh -L 127.0.0.1:8080:localhost:80 example.com
```

Copy progress bars can be misleading with slow flash drives, thanks to the way Linux buffers filesystem contents.

The good news is that it is possible to edit the label on a filesystem without affecting its contents, but the tool you need to use depends on the filesystem on the drive. Flash drives are normally formatted with one of the Windows filesystems: FAT for drives up to 32GB, exFAT or NTFS for those over – although some larger USB sticks still use FAT, despite the limitation of not being able to store files larger than 4GB. So, first you need to determine the filesystem. With some desktops, you can right-click the icon and find this info in the Properties window. Alternatively, with any distro or desktop you can open a terminal and run:

```
$ findmnt --list
```

This lists all the mounted filesystems on your computer, along with their filesystem. Make a note of the filesystem and the device, under the FSTYPE and SOURCE columns respectively. The program you want will be one of *fatlabel*, *exfatlabel* or *ntfslabel*, for FAT, exFAT and NTFS filesystems respectively. You probably already have these installed, but if not the relevant packages are:

dosfstools, **exfatprogs** and **ntfs3g**, all of which can be installed in the normal way. To see the existing label, but the program with just a device name, for example, use:

```
$ exfatlabel /dev/sdb1
```

This prints the current label, if any. To change the label, run:

```
$ exfatlabel /dev/sdb1 COOLSTUFF
```

The label does not have to be upper case, but Windows systems expect it to be. The **-r** option with *fatlabel* can be used to remove a label. The other two programs have no such option, but you can set it to an empty string with:

```
$ exfatlabel /dev/sdb1 ""
```

There are also similar commands for Linux filesystems, *e2label* being the one

for ext2/3/4 filesystems, but these are rarely used on flash drives.

Finished not finished

When I copy a large file to a USB 3.0 pen drive in Linux Mint 21.3 or 22, a window opens telling me what is left to copy, but after a few minutes it finishes copying and the window closes. I think the file has been copied to the drive and I click on the tab to disconnect it, but a window appears telling me not to remove the drive because the writing is still in progress. How is this possible? When I copy large files to an external SSD drive, I don't have any problems. I tried Pop!_OS 22.04 – when it starts copying the file to the USB 3.0 pen drive, a window tells me how the copy is going and when it finishes copying the window closes. I go to disconnect the drive and I don't have any problems. Is there a solution for Mint?

Phoebe Noble

A Linux buffers files in memory so that your system performance is not degraded by slow drives. As far as the program doing the copying is concerned, the copy has completed and it can get on with something else. However, when copying to a relatively slow device, and this includes flash drives (the flash is slower than the USB 3.0 interface), the data you copied is still in the buffer and being written to the drive in the background by the kernel. This is why you are not allowed to safely remove the device, and why your data would be corrupted if you simply unplugged it.

You can see this in action if you copy a number of large files to a flash drive using:

```
$ cp -v file1 file2... /media/pendrive
```

The first file appears to copy quickly but then there is a long delay for the

subsequent files as they wait for the buffers to be cleared.

One solution is to mount the device with the **sync** option. This causes the copying process to not return until the data has actually been written to the device. This is probably how Pop!_OS is handling it. You can tell this by running *mount* in a terminal and looking for **sync** among the options for the pen drive. If this is what you want, you can change the default options when mounting removable disks in */etc/udisks2/mount_options.conf*. Create the file if it does not exist and add:

```
[defaults]
```

```
defaults=sync
```

```
allow=exec,noexec,nodev,nosuid,atime,noatime,ro,rw,sync,dirsync
```

However, this can cause excessive writes to the flash drive, particularly if it is using a FAT filesystem, which can reduce the lifespan of the drive. So, only use this 'solution' when there is no alternative.

Another option is to run the *sync* command in a terminal. This tells the kernel to flush all buffers to their destinations and does not return until it has done so.

The reason you do not see this when copying to a SSD is because those drives are so much faster; write speeds are not so far removed from the read speed of the source, if not higher. Also, you do not usually try to remove an SSD immediately after copying to it. If you want fast flash-based external storage, consider using an SSD drive with a USB adaptor. You can buy fat and compact cases for less than a cheap USB drive. **LXF**

GET HELP NOW!

We'd love to try to answer any questions you send to answers@linuxformat.com, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (although many suspect Neil part dolphin), so it's important that you include as much information as you can. If something works on one distro but not another, tell us. If you get an error message, please tell us the exact message and precisely what you did to invoke it.

If you have, or suspect, a hardware problem, let us know about the hardware. Consider installing *hardinfo* or *lshw*. These programs list the hardware on your machine, so send us their output. If you're unwilling, or unable, to install these, run the following commands in a root terminal and send us the **system.txt** file, too:

```
uname -a > system.txt
```

```
lspci >> system.txt
```

```
lspci -vv >> system.txt
```

Mailserver

WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email letters@linuxformat.com.

Even Ubuntu is somewhat worried we're still using Ubuntu 17.04.

Answering answers

A robust and simple solution to the **LXF315** question entitled *Deep Search* in *Answers* lies in the single graphical application, available in the repos: *Recoll*.

It was reviewed in *Hot Picks* in a past edition of your top-class magazine (**LXF311**). You can build an index of every word in every document in a file hierarchy. With a user's *Thunderbird* email, it will also index every word in every email and all non-binary attachments, including PDFs. The resultant database can be searched for as simple an expression as a single word or for multiple words.

I regularly search a *Thunderbird* profile of 250MB, and it can find a word or expression in a flash. If it is in an email or in an attachment, I can open that file there and then. Left as installed, *Recoll* looks to the folder `~/.recoll` and defaults to indexing the `~` (**home**) directory. A simple variation on the default is to create separate databases, each for an index of a subset of folders. Want to index a *Thunderbird* installation – emails and attachments? Do the following...

- Create a **Recoll** folder for your Thunderbird profile:
`$mkdir -p ~/.Recoll/e-mail/zguvjznm.default-release`
- Launch *Recoll* and point it to the folder:
`$recoll -c ~/.Recoll/e-mail/zguvjznm.default-release/`
- *Recoll* prompts you to create an indexing configuration and indexing schedule – in this instance, configure for just the folder:
`~/thuderbird/zguvjznm.default-release/`

Linux Lite: designed for first-time Linux users coming from Windows. It's not just us saying that!



Et voila! (The effect of the Paris Olympics lingers still...)

- Repeat for any other folder of interest:
`$mkdir -p ~/.Recoll/Documents`

Multiple indexes can be created using this method. Just remember to launch *Recoll* in a terminal:

```
$recoll -c </path/to/index/>
```

This replaces the default *Recoll* action of looking to the folder `~/.recoll` for its index.

Thanks to all your authors and contributors for a truly great magazine.

Tony McNamara, Western Australia

Neil says...

Thanks for the pointers. We should read this *Linux Format* – it sounds helpful!

Linux lies!

You reviewed Linux Lite 7 in **LXF318**, stating that it was recommended for those moving from Windows to Linux.

Having moved over 50 laptop users and nearly as many desktop users, I'd disagree with that statement. I use Linux Mint with the Cinnamon desktop. Having here someone whose Windows machine was wiped by MS in 2018, used Windows one day and Linux since then, Linux Lite is definitely not what I would have given her.

Mint has a logical layout and is simple to see what is where. It is similar to Windows 7 and 10 but more logical. Linux Lite is not like Windows. Linux Lite might come from New Zealand, as I do, but that doesn't mean it does as suggested.

Sandy Ferguson

Software Updater



Software updates are no longer provided for Ubuntu 17.04.

To stay secure, you should upgrade to Ubuntu 18.04.6 LTS.

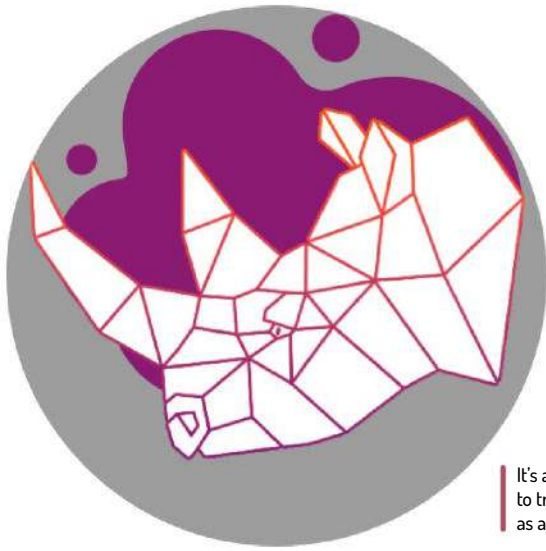
Settings...

Upgrade...

OK

Helpdex





It's a fine distro to try, just not as a beginner.

Neil says...

To give you full credit, Linux Mint did win our beginners' distributions *Roundup* this month (see page 26). Having said that, I'm looking at the Linux Lite website right now and I'm reading: "The free, easy-to-use operating system." Also: "A familiar environment anyone can use." That sounds as though it's targeting first-time Linux users who are used to Windows. Even if you think it shouldn't be, that's probably going to be the assumption most people have – and, to be honest, so do we!

Tarnished Chrome

I am running Ubuntu 17.04. Would the latest version of the *Chrome* web browser run on my computer? If so, how would I install it?

Ian Learmonth

Neil says...

That's a rather odd version of Ubuntu to choose to stick with – it's not even an LTS (long-term support) release, such as 16.04 or 18.04, which would still be supported with security updates. The way to get *Google Chrome* on your computer is to update to a still-supported version of Ubuntu, such as those mentioned – unfortunately, *Chrome* won't now install on Ubuntu 17.04 due to unavailable dependencies.

Rhino rush

I read your recent article about Linux Rhino (*LXF311*) and I was wondering whether it would be suitable for a beginner?

Jed Champion

» LETTER OF THE MONTH

Super safes

I buy your magazine every month, and I must say that I enjoy it. But, not to be complacent, I have to confess that I do not understand much of it. But I do get satisfaction from reading about Linux. I mention this, not as a complaint, simply to highlight the fact that my question may be regarded as not relevant, or not deserving to be answered.

My spreadsheet, finances and other private stuff are stored on a laptop, which is not connected to the internet. This, I assume, is the safest place for this type of data to be.

Data will be entered from receipts into the spreadsheet.

Occasionally, I go to the bank and ask for a bank statement for the past week, and I compare the figures to see that they tally. If there is an issue, I take the following steps: using another laptop that already has Mint, I load *Lubuntu*, without installing. I believe this is called a live ISO. I log into my bank and check the details on my bank account, and compare that to my spreadsheet on the computer, to seek out the irregularities. From YouTube, I am led to believe that this is a safe way to bank online.

For my purposes, this is ideal. However, procedures about basic online banking are lacking. Is it safe for me to continue with this method?

John Graham



We'll be looking at security-conscious distros like Qubes soon.

Neil says...

I'll say it sounds somewhat over the top, but it's as safe as the systems allow. I can understand using a live distro as it ensures a clean system – presuming the live image hasn't been tainted, if we're being paranoid, but you should be checking this with the provided hash, too. These days, the real security risk is the human using this all – you. Scammers have shifted to trying to scare people into transferring money and can even use AI-generated audio to pretend to be family members in trouble requesting funds.

Neil says...

I would say no – you would be much better off just going with Ubuntu 24.04 or Linux Mint – see the *Roundup* in this very issue (page 26) for more pointers! That's not to say that Rhino is difficult to use, but it is based on the development build of Ubuntu, which potentially could throw up issues and bugs down the line. **LXF**



WHAT SEEMED LIKE A PERFECTLY VALID QUESTION ...

shane_collinge@yahoo.com

SUBSCRIBE Save money today!

SUBSCRIBE TO LINUX FORMAT MAGAZINE SAVE 50%*

www.magazinesdirect.com/LIN/C75J

Call **0330 333 1113**
and quote **C75J**

NEW!

Digital access to
130+ issues when
you subscribe
to print!**



Save money today! **SUBSCRIBE**



OUTSIDE THE UK?
Turn to page 61 for more great subscriber deals!

» **NEW:** Improved digital back-issue access!**



Get **instant access** on your browser, or Android and Apple iOS devices back to issue 181 (March 2014) with 1,000s of tutorials, interviews, features and reviews.

» CHOOSE YOUR PACKAGE!

Keep your subscription rolling!

Subscribe today and save 50%!* + All-new digital access**

PRINT EDITION



Only
£21.09

for 6 months

6 months of *Linux Format* delivered for £21.09!

DIGITAL EDITION



Only
£22.71

for 6 months

6 months of digital *Linux Format* for £22.71!

SAVE!
50%*

Terms and conditions: **Offer closes 15th October 2024.** Please allow up to 6 weeks for the delivery of your first subscription issue (up to 8 weeks overseas). The full subscription rate includes postage and packaging. *Savings are based on the cover price. Payment is non-refundable after the 14-day cancellation period. **Access to the digital library will end with your subscription on print. For a digital subscription, once terminated you only keep the issues you have paid for up to that point. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries and overseas rates please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 8:30am-7pm, Saturday 10am-3pm UK time (excluding bank holidays) or email: help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

EXPLORE THE BEST LINUX & OPEN SOURCE SOFTWARE

From the best distros to system security, utilities and fun applications,
we've rounded up everything you could ever need



Ordering is easy. Go online at:

magazinesdirect.com

Or get it from selected supermarkets & newsagents

GnuCash

The words “free accounting” are music to **Rob Clymo**’s freelance ears.

IN BRIEF

GnuCash is personal and small-business financial-accounting software that’s been well maintained since 2001. More critical types might accuse it of being a little dated, but the features and functions do all work once you’ve picked your way through the quite time-consuming setup and configuration stages. It’s an ideal option for someone running a small business who needs to keep a firm grasp on cashflow.

GnuCash is a completely free open source accounting package that is impressive to say the least. Although it has more of an accounts-style edge, the program lets you tackle all manner of financial chores, including overall management of your money.

What that means is that for no financial outlay, you can quickly get access to a system that enables you to set up and manage things such as invoicing and payments, customer and vendor management, as well as budgeting. In fact, due to the way *GnuCash* works, you’ll find that it becomes a hugely useful tool for managing budgets, especially so if you’re running a small business.

Adding to the appeal is its comprehensive range of reporting tools, which any small business owner will find hugely useful. Given that *GnuCash* comes with so many complex features and functions, it’s hard not to feel very happy with the software once you’ve installed it. Some paid-for programs don’t offer as much and, indeed, a few don’t work as well either.

Called to account

While it can be used as a basic money management tool, *GnuCash* is actually at its best when it’s being utilised for more accountancy-based tasks. It covers a raft of those, including accounts payable and receivable chores, managing customer and vendor requirements, as well as producing lots of reporting, so you can easily keep tabs on your money matters.

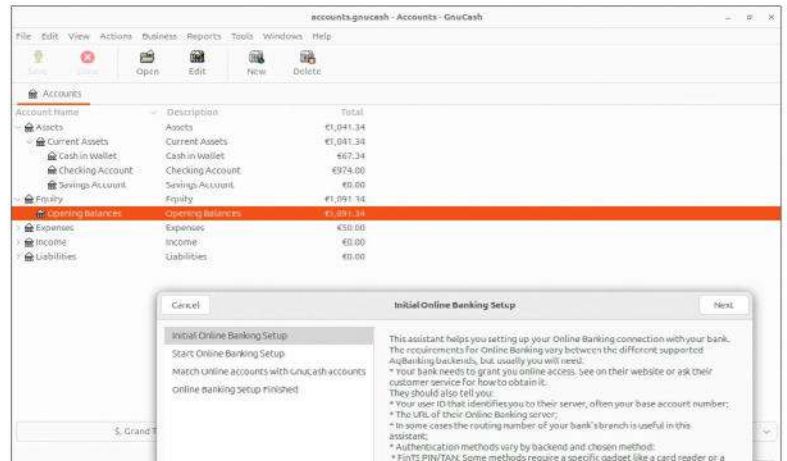
The budgeting aspect of the software kicks in nicely by tying all of these features together, and there are pretty sophisticated menu options for setting this up.

Indeed, if you’ve loaded a lot of financial information into the system, it rolls along very commendably, with nothing to make you live in fear of losing a year’s worth of accounts or something similarly cataclysmic.

GnuCash is everything you’d expect from open source software. That is to say it comes with lots of features and functionality, but also has a few rough edges. Once you get into the configuring stage, this can cause one or two headaches. The real bonus is the assistant that helps get you up and running, while another practical touch is that *Quicken* users can also import QIF files.

We also found the default business hierarchy a real boon as it sets out a stall for you and your business, which can then be tailored to suit your own needs.

GnuCash does have a reasonable level of support options available, and you may well require their



The LXF cash reserves are looking in dire need of topping up!

services due to a few quirks within the software. To get you started, there’s a comprehensive online help manual and that takes you on a deep dive into its many different facets.

This is particularly useful when you’re working through the initial setup stages, because these areas of the program can be a little tricky to get your head around if you’re not familiar with its nuances. Adding to the interest is the fact the user interface looks a little old school with its slightly dated feel.

We also found the tutorial guides quite useful for the same reasons, while the *GnuCash* wiki content is also a very practical way to get primed about issues you’re stuck on, as are the old faithful YouTube videos.

Overall, it’s really quite hard to fault software like *GnuCash* because it has been put together so well and, ultimately, doesn’t cost you anything in order to use it. Even better is the fact that *GnuCash* does such a good job, especially if you’re running your own business, where you need a little bit of a helping hand to keep your finances in order, but lack the budget to head down a paid-for software route. **LXF**

VERDICT

DEVELOPER: The GnuCash Project

WEB: www.gnucash.org

LICENCE: GPL v3

FEATURES 7/10

PERFORMANCE 7/10

EASE OF USE 6/10

DOCUMENTATION 8/10

If you’re looking for a no-cost alternative to the many paid-for money management programs, *GnuCash* is very effective. It gets the job done in an efficient no-frills fashion.

» **Rating 7/10**

Vanilla OS 2 Orchid

Is Vanilla OS a darling bud of May or just an awkward orchid?

Nate Drake is on the case.

IN BRIEF

As an immutable operating system, Vanilla OS 2 promises better security and effortless updates. While we struggled with setup, once installed the OS ran perfectly and installing extra software was simple.

SPECS

CPU: 1GHz
Mem: 4GB (8GB ideal)
HDD: 50GB
Builds: x86_64

Vanilla OS first blossomed in February 2023. The first version, code-named Kinetic, was based on Ubuntu. The second fixed release (code name Orchid) is now based on Debian Sid and uses the Gnome desktop environment.

Vanilla is immutable and uses ABRoot for atomic transactions; in other words, core system components are locked down against changes by third parties.

Upon proceeding to download, we discovered that the distro is available in a 1.6GB ZIP archive, which when decompressed revealed a 1.7GB ISO.

To quote the project website, this distro apparently emphasizes “simplicity, cleanliness and freedom”. The first of these wasn’t very present when trying to install Vanilla in a VM, as UEFI support is required. After enabling this feature in *VirtualBox* and performing a fresh install, the OS also failed to boot, despite the main website claiming Vanilla OS is compatible with virtual machines.

After we did manage to get Vanilla working, we found the setup to be reasonably intuitive, though the fact that keyboard layouts are listed alphabetically meant it took some scrolling to get to ‘English’ options.

Post-install, a wizard launches to “take care of everything”. In this case “everything” encompasses choosing the language and keyboard layout again, as well as creating a user account and device name.

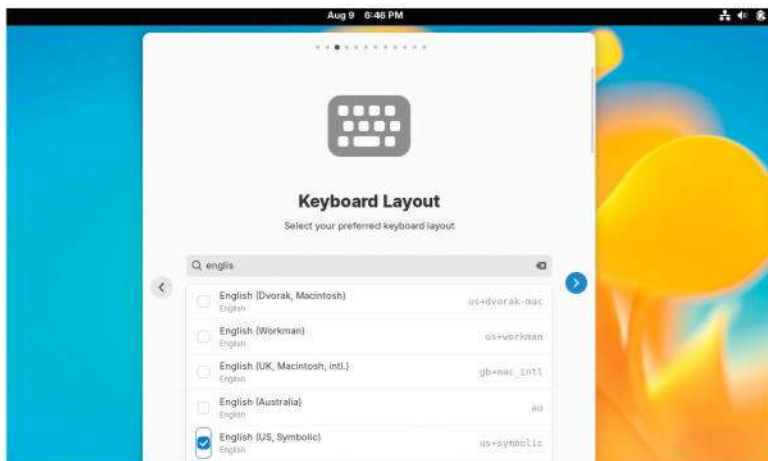
Vanilla OS 2 next prompts you to choose which apps to install. By default, core Gnome apps and web browsers are selected, though you can also install *LibreOffice* and common utilities like *Sound Recorder*.

The system now reminds you that it is “immutable and atomic”, then reboots to a GRUB prompt offering a choice of Current or Previous state.

After choosing the former, a login screen appears where you can set your account password. This in turn leads to the *First Setup App* launching, which starts the process of “finalizing” the install, including downloading and installing requested programs like the Flatpak versions of Gnome core apps.

All of the above, including aborted setups, meant it took us over two hours from downloading the ISO to setting up a viable desktop. We can cut Vanilla OS 2 some slack for being atomic and immutable but other distros, such as Fedora Silverblue and Endless OS, that also meet these criteria are much faster to set up.

This distro also clearly isn’t for legacy machines. The listed system requirements on the website help section



We found setup extremely tricky and time-consuming on our virtual machine but readers have more joy on bare metal.

recommend 8GB of RAM and 50GB of free disk space for installation. Our VM’s virtual disk came to just over 19GB post-install.

Unlike other immutable operating systems we’ve reviewed, Vanilla OS uses Btrfs on system partitions.

On the plus side, like most immutable systems, Vanilla OS 2 handles updates automatically. By default, they’re applied weekly and the OS checks for Smart Updates whenever the computer is idle. The welcome guide also claims Vanilla OS supports Android APKs. *Gnome Software* also comes bundled, which we used to install the Flatpak version of *Mines* in seconds.

Special mention should also go to the *Vanilla OS Tour* app, which provides an overview of the distro’s features. (This is where we learned about how it manages updates.) However, some of the listed perks, such as integrating email accounts, are available through Gnome and are not Vanilla-specific.

Talking about Gnome, we also fired up *System Monitor* to find that Vanilla OS 2 used around 2GB of RAM and only around 3% of CPU resources at rest. **LXF**

VERDICT

DEVELOPER: Fabricators et al

WEB: <https://vanillaos.org>

LICENCE: GPL v3

FEATURES	8/10	EASE OF USE	4/10
PERFORMANCE	7/10	DOCUMENTATION	7/10

A long and frustrating setup aside, Vanilla OS 2 lives up to its promise of being clean and immutable.

» **Rating 6/10**

Emmabuntüs DE 5 1.02

Nate Drake searches frantically for his umlaut key in evaluating this ethical, lightweight and accessible French distro.

IN BRIEF

Emmabuntüs likely contains every software package you'll ever need, as well as a choice of lightweight desktop environments. The developers have also introduced better accessibility features for the visually impaired.

SPECS

CPU: 2GHz
Mem: 1GB
HDD: 40GB
Buils: i386, AMD64

This distro's name isn't a bastardised homage to Baby Spice but a portmanteau of its origins. The Emmabuntüs Collective is inspired by the Emmaus international solidarity movement, which focuses on combating poverty. Although not officially recognised by the movement, the collective does a great deal of humanitarian work in refurbishing old PCs (complete with this lightweight Linux distro), then donating them to charitable causes in its home country of France as well as Africa.

The other part of the name reflects the fact that the OS used to be based on Ubuntu, though the development team has since switched to Debian Stable – in this case, Debian 12.6 Bookworm.

True to the aim to support older PCs, the latest version is available for both 32-bit and 64-bit systems.

The main site describes Emmabuntüs as an “all-in-one” Linux distribution. This is one perk of the OS, as if you have the patience to download the 3.9GB ISO, it comes with a whole truckload of pre-installed apps.

On the plus side, this means minimal downloads of extra packages during install and use. However, when we booted the live version into a virtual machine, we couldn't help but feel some software is redundant; for instance, it contains both *AbiWord* and *LibreOffice*, *Clementine* and *VLC*, *Falkon* and *Firefox ESR*.

This said, Emmabuntüs is highly customisable. On first boot, you can opt for the graphical installer, which is virtually identical to Debian's. If you instead boot into the live environment, as we did, setup is done via the *Calamares* installer. Users are also offered OEM mode to configure the distro to their needs prior to install.

Emmabuntüs defaults to Xfce but also comes with the LXQt desktop environment. The welcome screen also offers options to tweak the default wallpaper as well as configure the Cairo dock.

The final dock setup option offers three modes: All, to list all installed software; Simple, for newbies; and Basic. These determine which applications are listed, so having complained about the huge number of overlapping packages it's a relief to know users aren't obliged to see them all.

Upon login, Emmabuntüs defaults to a welcome window that contains helpful links to website tutorials on how to set up the OS as well as how to refurbish your computer.

The distro also bundles *Gnome Software*, which we used to install *System Monitor*. When using Xfce and



The OS offers a choice of both the Xfce and LXQt desktop environments. Users can also configure cairo-dock.

LXQt, RAM usage came to just over 1GB, which is in keeping with the listed system specs and tutorials.

Despite choosing to install non-free packages like Microsoft fonts and codecs, Emmabuntüs had a very light install footprint of just over 8GB. Special mention should also go to a feature we hadn't encountered before post-install: having selected US English as the default language, Emmabuntüs offered to remove all other pre-installed languages to free up space.

The latest version of Emmabuntüs also places an emphasis on accessibility features, particularly for the visually impaired. Said features can now be accessed via a specific window, which includes configuration options for the *Orca* screen reader. From here, users can also tinker with *Compiz* settings such as zooming and full screen.

The OS now also includes *NatBraille* for generating Braille books, as well as *Daisy-player* and *eBook-speaker* for playing audio books. The default calculator has been replaced with *gnome-calculator*. The distro also now supports accessibility shortcuts. **LXF**

VERDICT

DEVELOPER: Collectif Emmabuntüs

WEB: <https://emmabuntus.org>

LICENCE: Mainly GPL

FEATURES	9/10	EASE OF USE	8/10
PERFORMANCE	8/10	DOCUMENTATION	9/10

Ethical, lightweight and with excellent accessibility features. Our only complaint is the number of redundant packages.

» **Rating 9/10**

Sleeper OS Bee v2

Nate Drake fires up his laptop to discover whether Sleeper OS is the perfect dream or every Linux user’s nightmare.

IN BRIEF

The default version of Sleeper OS is easy to set up and boots fast. It’s let down by a dubious selection of pre-installed apps, patchy documentation and frequent references to MX/antiX.

SPECS

- CPU: 1GHz
- Mem: 512MB (1GB recommended)
- HDD: 5GB
- Builds: x86_64

Sleeper OS is a new Linux distribution. So new, in fact, that DistroWatch has yet to add it to its comprehensive online database. In its short lifetime, the OS has gone from being based on Rocky Linux to “embracing an MX Linux base”. MX itself is based on Debian stable, while using components of antiX.

This means that, like antiX, Sleeper OS doesn’t use systemd. While the project doesn’t have a dedicated website, its SourceForge page does have a basic wiki and the code for a site. This is where we learned that Sleeper OS is designed for “artists and hobbyists”. This may account for the flamboyant names of the distro’s alternative spins, including Snow White Gnome Desktop, OK KDE Desktop, Ratolí Xfce4 and Mountain Cinnamon Desktop.

We opted for the default version, which uses Trinity. After booting the 1.4GB ISO of the latest Sleeper OS Bee v2, it showed its roots by offering to boot “antiX”.

On the first boot of the live desktop, we were initially stymied by a password request but the documentation prompted us to enter user:live, password:live.

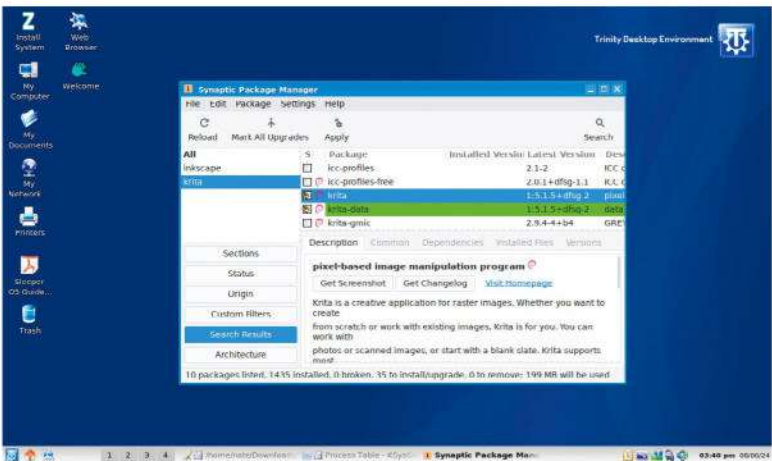
We also noted a Sleeper OS Guide PDF, which we opened to discover the system requirements for the default version. Here we also learned that the developers pride themselves on the distro being lightweight and fast. Eager to put this to the test, we fired up the default installer. It was easy to select the main disk via a drop-down menu and begin setup, but we couldn’t see any option for full disk encryption.

If you’ve made changes in the live environment there’s also an option to preserve these post-install. Although we didn’t check this, we noted upon reboot that the Install System shortcut was still present on the desktop.

We next decided to test Sleeper OS’s artistic credentials by examining the default apps. Support for painting/imaging is handled by Chalk, while users can also work with scalable graphics via Carbon 14. The distro also comes with multiple audio players such as XMSS, JuK and Noatun. We also noted a CD ripper (KAudio Creator) and CD player (KsCD), but couldn’t help but feel these are a little redundant in 2024.

It’s also hardly in line with the project’s stated goal on SourceForge of minimising bloatware. This said, there was no sign of Inkscape or Krita, despite them being described as bundled in the project wiki.

In view of this, we fired up the LPKG software manager, originally designed for LOC-OS Linux. This



Although the LPKG package manager has a limited selection, Sleeper OS also bundles Synaptic for installing packages from the Debian repositories.

offered a limited number of packages, which we used to install the Leafpad text editor. We had more joy with the terminal, which we used to install Tetris clone Tint. When we tried to do the same for Inkscape and Krita, we were told the package wasn’t found.

Sleeper OS does, however, come with KSysGuard for monitoring system performance. With no other running programs, the default version of Sleeper OS uses around 1.3GB of virtual memory, a far cry from the stated minimum of 512MB.

After firing up Firefox to put Sleeper OS further through its paces, we noted that the browser defaults to the MX Linux homepage. The bundled bookmarks also link to MX and antiX Linux user forums.

After downloading a 4K desktop background, we opened it in Chalk. The editor has a respectable number of basic features, such as cropping, as well as adding shapes and texts. Users can also add certain filters, like a Gaussian blur. Still, we couldn’t help but wonder why a distro aimed at artists wouldn’t include a more comprehensive image editor, such as GIMP. **LXF**

VERDICT

DEVELOPER: Alejitao
WEB: <https://sleeperos.sourceforge.io>
LICENCE: GPL v3

FEATURES	5/10	EASE OF USE	7/10
PERFORMANCE	7/10	DOCUMENTATION	4/10

There’s nothing functionally wrong about Sleeper OS but it feels half-finished. It’s also unclear what it has to offer artists.

» Rating 6/10

Super Grub2 2.06s4

Is it a bird? Is it a plane? No, it's Super Grub! **Nate Drake** shares how it worms its way into unbootable systems.

IN BRIEF

Super Grub2 Disk 2.06s4 boots broken OSes with ease but we want detected systems displayed at the top of the list. A repair option for corrupted/boot partitions would be great, too.

SPECS

CPU: 1GHz
Mem: 512MB
HDD: N/A
Builds: i386, x86_64, i386-efi, x86_64-efi

The words 'super' and 'grub' seldom go together but the title is well earned in the case of this live CD, as it allows users to boot into virtually any operating system.

The moniker Grub2 derives from the fact that given GRUB2 was a complete rewrite of the GRUB bootloader, Super Grub2 Disk includes changes to GRUB Legacy. Another crucial change to this version of the OS is that it cannot write to the system partition or MBR at all. Naturally, this means it's incapable of repairing corrupted drives but it can rescue booting of a number of systems.

On heading to the main site, you are offered the choice between USB-writable IMG files or traditional ISOs. These are broken down further into 'Oldie' images for 32-bit and 64-bit machines as well as downloads for more modern UEFI systems.

In the case of the ISOs, we found on average these were around 20MB. This is unsurprising as Super Grub2 Disk doesn't make any pretence of being an operating system, given there's no graphical environment and it can't be installed to disk. The developers state that if you're unsure, choose the hybrid version of Super Grub2 Disk as we did.

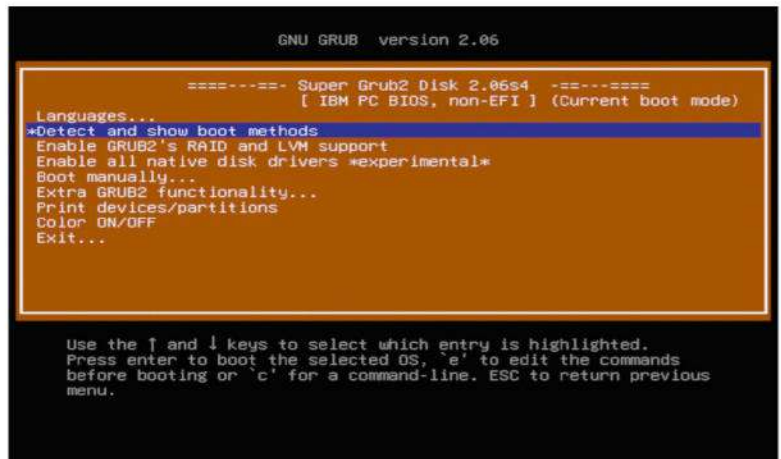
From reading the project wiki, it's clear that some effort has gone into expanding the distro's repertoire since its last stable release in 2019. Btrfs file systems are now supported.

Super Grub2 Disk also now offers operating system specific options for EFI, FreeBSD, FreeDOS, Linux, Mac OS X, MS-DOS, Windows 98, Windows NT, Windows Vista (and newer), GNU/Hurd and ReactOS.

Upon booting, users are presented with a navigable text interface. Power users most likely will jump to Print Devices/Partitions to see a readout of all detected drives. This, in itself, won't help your OS boot, which is why the second menu option (Detect And Show Boot Methods) is preferable.

Once selected, Super Grub2 Disk simply tries to detect relevant partitions for all the operating systems it supports. This usually means you have to navigate past all the various OSes that are 'not detected' until you come across your own. We couldn't help but feel it would make more sense to automatically display detected operating systems at the top of the list. Nevertheless, it's very simple to use the arrow keys and Enter to boot the selected OS.

The main menu also lists the option to view Extra GRUB2 Functionality. From here you can search



The main menu contains options to automatically detect unbootable operating systems, as well as extract entries from grub.cfg files.

external CDs and floppy disks, as well as attempt to mount encrypted LUKS and geli volumes.

The Boot Manually section also contains an option to extract relevant information from **grub.cfg** files, as well as boot various operating systems when the automatic detection feature isn't working.

Revelling in his capacity for self-sabotage, Nate fired up his Ubuntu VM and erased the **/boot** partition via *Gnome Disks*. On restarting, this caused the system to load to a blinking white cursor on a black screen.

Upon loading Super Grub2 Disk, he selected Detect And Show Boot Methods to see the Linux kernel file in the **/boot** directory – once selected, it booted Ubuntu.

Special mention should also go to the developers for bucking the Anglo-centric trend of software. Super Grub2 Disk supports numerous languages including French, Spanish, Chinese, Russian and Polish.

The release notes state that Windows 10 and 11 are supported, but when we tried to boot ISOs into a Windows 11 VM, Super Grub2 Disk failed to load. This won't weigh heavily on the minds of Linux users... **LXF**

VERDICT

DEVELOPER: Adrian15 et al
WEB: www.supergrubdisk.org
LICENCE: GPL v3

FEATURES	8/10	EASE OF USE	8/10
PERFORMANCE	9/10	DOCUMENTATION	6/10

Super Grub2 Disk is a good example of a distro that does one thing very well: recovering unbootable operating systems.

» **Rating 8/10**

Schim

Management is thrilled that **Jody Macgregor** likes working in the shadows.

SPECS

OS: Ubuntu
64-bit
CPU: Intel
Core 2 Quad
Mem: 2GB
HDD: 2GB
GPU: GeForce
GTX 650,
1GB VRAM

Back in 2007 or so there was a real trend for indie puzzle-platformers, usually about a sad boy who had to go from the left of the screen to the right for some reason. Although it's not a 2D platformer, *Schim* reminds us of that era, when indie games took a single mechanic then expressed it through a different iteration in each level while squeezing as much emotional heft as possible out of storytelling on a budget.

In *Schim* you play a shadow creature that can't exist in the light and has to hop from one puddle of darkness to the next, like you're playing goth *Frogger*. With your eyeballs poking up out of the shadows as you swim around in them, there's a definite frogginess to the main character. Occasionally, you see other shadow frogs hopping around in other shadows, with gigantic ones lurking in the darkness beneath trucks and the like. You're a more ordinary shadow frog, though, who lives in the shadow cast by a single person – until the moment you're separated, beginning a quest to find your way back to the sad boy whose silhouette you call home.

Sometimes hopping is a matter of timing. You wait for a car or a box on a conveyor belt to pass so you can jump into its shadow, and then on to the next. You can also interact with objects and beings whose shadow you're in, pressing buttons to open doors or raising a forklift's forks to lengthen its shadow so you can rush to the edge and leap before they lower. This is *Schim* at its best, when you make a duck honk at a cat or turn on street lamps to make new havens you can navigate. The latter is necessary because there's no refuge in the actual darkness of night, which is as deadly as sunlight. It's not the absence of light you need, but the shadows cast by it.

The city – and a slice of countryside – you cross is Dutch, as are *Schim*'s creators. Some of the maps could be almost any videogame city with cars driving on the right, but occasionally you see a parked bicycle dangling



Look at all those bikes and not a bike thief in sight – this isn't Britain.

over the side of a bridge it's chained to, or an unusually clean park, and realise you're in continental Europe.

At the start of each chapter, you're treated to a pan over the map, but you don't need to plan your path, as it's clear where you need to go. Painfully so. Where you might expect the interactions to build into brain-bending puzzles or challenging feats of jumping, they never do.

Schim's ease suggests it's designed for relaxation rather than intensity, yet those long pans across the level don't make us feel chilled out at all. They make us feel exhausted, each one showing where the sad boy we're following has gone, a position we know he'll have moved on from by the time we get there. Every level ends in a tedious 'princess in another castle' tease for the next.

We were dealt a real blow by that feeling when going back to *Schim* after a break to find it had lost an entire level of progress. Replaying and trying different things revealed that, beyond a few collectibles hidden down dead ends, there's nothing to *Schim* beneath the surface.

Condensed down to its best moments, *Schim* would have been an impressive student game, but stretched out to the length demanded by full-price boutique indie-label games, it becomes a chore. The inventive puzzle-platformers of 2007 cast a long shadow, and *Schim* doesn't do enough to escape from beneath them. **LXF**

Dutch railway stations are renowned for their lack of lighting.



VERDICT

DEVELOPER: Extra Nice, Playism

WEB: <https://schimgame.com>

PRICE: £21

GRAPHICS 6/10

GAMEPLAY 6/10

LONGEVITY 5/10

VALUE 5/10

Not difficult enough to be a challenge, but not painless enough to be relaxing.

» **Rating 6/10**

YOUR DIGITAL ISSUE ACCESS

Linux Format print subscribers can now access* digital back issues two ways! Who's a lucky bunch of readers?!

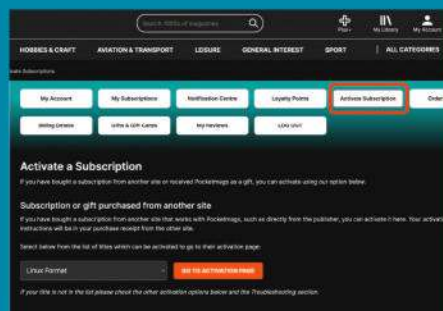
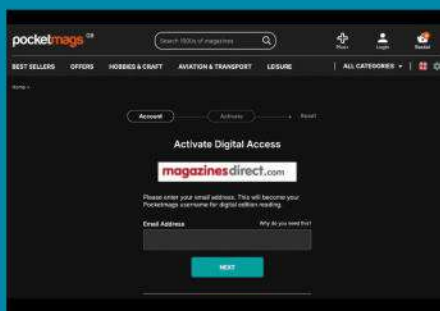
» As a big thank you for subscribing to *Linux Format*, we've always offered digital access to past issues. It seems The Management thought it was a good idea and has rolled out a company-wide system using the Pocketmags service!

This means there are now two ways of accessing LXF back issues for subscribers. The new Pocketmags service goes back to March 2014, including over 140 full issues, while our older PDF archive goes back to LXF66 (a crazy 8-bit filling 254 issues), though issues before LXF120 have had covers and ads stripped. We've outlined how to access both below. Enjoy!

*terms and conditions apply see page 17 for full details.



FANCY NEW POCKETMAGS

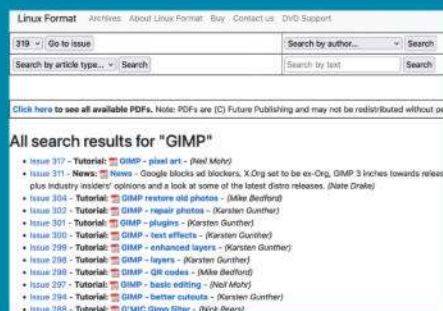
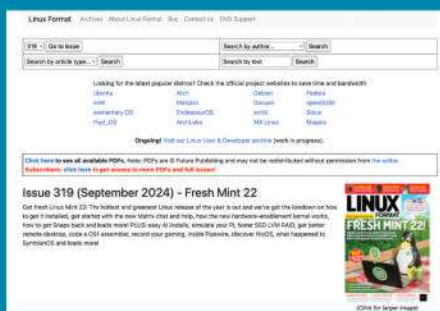


1 Activate your access
Head to www.magazinesdirect.com/digital-instructions/ and click the Start Reading link that'll transfer you to the Pocketmags activation page.

2 Create an account
Enter the email that was used to create the subscription. You'll be prompted to create a Pocketmags account so you can access the library.

3 Activate your access
On the Activate A Subscription page, select Linux Format from the pull-down menu and click on Go To Activation Page. Select Library – happy browsing!

THE CLASSIC LXF PDF ARCHIVE



1 Head to Linux Format
You heard the man – browse to www.linuxformat.com/archives using your favourite web browser. Where it says Subscribers: Click Here, do just that.

2 Use your subs number
Log in using your subscriber number and surname. You can find the number above your name and address on the address label.

3 Access your issues
Use the issue drop-down menu to access past and current issues. You can also search by author and keyword or article types.

Visit www.magazinesdirect.com/linux-format

Roundup

MX Linux » Ubuntu » Linux Mint
» Pop!_OS » Linux Lite



Michael Reed

is a Linux professional and freelance writer now. But once upon a time, he was a Linux beginner, too.

Beginners' distributions

Not everyone is a Linux veteran like **Michael Reed**, so he's assessing five distributions that could be a good choice for Linux newbs.

HOW WE TESTED...

For this *Roundup*, we considered things from the perspective of a Linux beginner. One of the first decisions we made was the choice of edition. In each case, we concentrated on the main edition of that distribution rather than confusing matters by delving into different flavours and derivatives.

On this occasion, we didn't carry out any benchmarking, but we kept an eye out to make sure that everything we did was performant. In our opinion, an 'old' desktop or laptop PC tends to be powerful enough to carry out all the basic computer tasks using a typical Linux distribution these days. If you're looking at equipping a really old machine with less than 4GB of RAM, lightweight distros are a separate category and one that we've covered in the past.

Hardware support is another area that we have to examine, but in each of these five cases, it looked like all typical eventualities would be covered, including installation of proprietary drivers.



This month, we're looking at five Linux distributions that are suitable for newcomers to Linux. A beginner in this context can mean a couple of things. On the one hand, it can mean a novice computer user. On the other, it might mean someone with good computer skills but who has no or little experience with Linux.

To this end, we're considering the experience for new users, all the way from installing the distribution themselves to carrying out the basic customisations and application installations. Beyond that, we have assessed the user interfaces from the same perspective and considered what

the support community is for each distro. When you add all this together, this means that we may have to underrate a good distro if it's less suitable for newbies. Things such as a good range of default applications are all the more important for this class of user.

Some of our scoring might feel unfair because we may have to give a low rating in areas where the requirements of a Linux desktop distribution are different for total newcomers to Linux. In particular, the final assessment of all of the distributions will lead to a lower ranking to what are, by standard measures, perfectly fine distributions for normal use.

Getting it installed

The installation process, from a beginner's perspective.

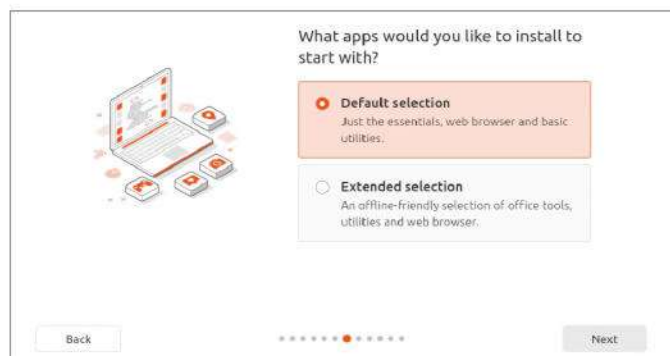
The latest Ubuntu (24.04) is also the most suitable for beginners, as it's a Long Term Service (LTS) version that will be supported for free until 2029. Installing an operating system requires some technical knowledge, but we think that Ubuntu's scheme of offering automatic partitioning along with other essentials couldn't be simplified any further.

The Ubuntu installer now has an accessibility section, which is good, but it doesn't have any explanation of what features such as Visual Alerts and Mouse Keys actually do. The installer checks for updates to itself and gives the option of installing a newer version. This requires a relaunch of the installer, and it might be a speed bump for new users. We like the choice between a basic installation or an extended one, and we chose the latter for extra apps. We also chose to install extra drivers and media codecs.

Linux Mint sticks to the fundamentals of the standard Ubuntu install but lacks the basic/full choice and accessibility options.

Pop!_OS uses an older version of the Ubuntu installer. It doesn't add any frills other than making encryption the default. As it is based on Ubuntu 22.04, there will be an upgrade soon.

Some distros feature a first-run dialog, but MX Linux presents you with this before you begin installation. Upon booting the MX



Many distros have done away with detailed package selection as part of the install process. We feel that Ubuntu strikes the right balance, with full and basic options.

Linux installation medium, you're greeted with an extensive welcome dialog packed with links to helpful resources. Like much of MX Linux, the installer is derived from the Mepis Linux one. It features a sidebar with a detailed description of options.

Linux Lite takes a similar tack by presenting a comprehensive help dialog before installation begins. There is some room for confusion as some options (such as setting a restore point) are more appropriate for a first-run dialog rather than preinstallation.

VERDICT

MX LINUX	9/10	POP!_OS	6/10
UBUNTU	9/10	LINUX LITE	8/10
LINUX MINT	7/10		

All good. Ubuntu sets the standard and has some sensible new features. MX Linux and Linux Lite offer extra help.

First-run experience

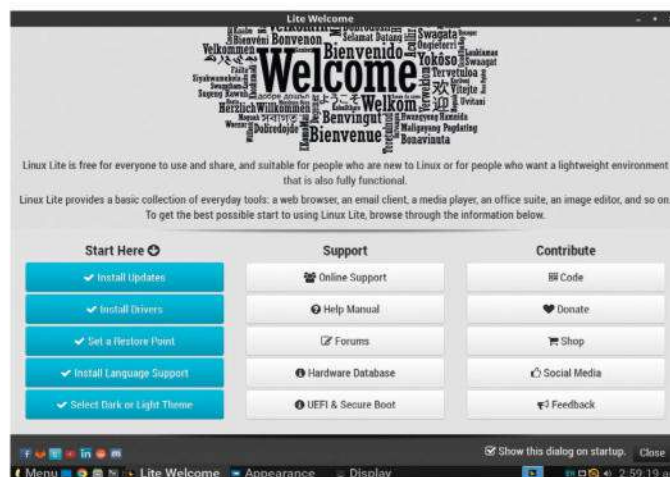
People new to Linux might need extra hand-holding at the start.

The Pop!_OS first-run dialog has some cosmetic options, such as the light/dark theme and the position of the dock. It also features an explanation of how the dock works – helpful to beginners. There is also an explanation of how gestures operate. The online services page of the dialog has a good selection of the mainstream services, such as Google.

Linux Mint provides a long scrolling list of options that cover areas such as software installation, the colour scheme, updates and driver installation, plus links to the forum, IRC channel and documentation. This first-run experience touches on most areas a new Linux user needs to know about.

When running Ubuntu for the first time a dialog offers an option to join Ubuntu Pro and leads to opening the App Center. All in all, it's not a detailed introductory sequence.

The animated MX Linux startup logo provides a bit of extra sheen to what is a highly customised distro overall. The welcome dialog that we first encountered during installation doubles as the first-run dialog as well. It's extensive without being overwhelming and hits every area that a newcomer is likely to need, such as the MX Linux forum and documentation.



The Linux Lite first-run dialog has links to documentation and support, including a row of social media icons. There are also maintenance options, such as Update.

Linux Lite also recycles the preinstallation help dialog as the first-run dialog. It shares the same shortcoming as MX Linux in that there are no options to set up online accounts, like Google.

VERDICT

MX LINUX	8/10	POP!_OS	7/10
UBUNTU	5/10	LINUX LITE	7/10
LINUX MINT	7/10		

Pop!_OS has some UI instructions and can set up online accounts. Linux Lite and MX Lite concentrate on support resources.

Getting around the desktop

Where you'll spend most of your time.

We like to work within a pleasant aesthetic, and a clearly laid-out, good-looking interface can pull new users into Linux and make them want to stick with it. Because we're considering things from a beginner's perspective, we're awarding more points to user interfaces that are easily discoverable and that resemble existing interfaces. It might seem unfair that we're, effectively, punishing innovation this time around.

We're taking these interfaces as we find them, but in most cases, if there is a feature that you really dislike (placement of panels and so on) you can change it – this is Linux, after all.

These days, we expect an app launcher to pop up via a hotkey and be fully searchable with a search that includes application descriptions, and we are happy to report that this is the case in each of these desktops. These aren't desktops that target low-spec machines, but on average hardware, we find them all to be snappy in operation.

MX Linux

7/10

Ubuntu

7/10

Xfce is the desktop environment of choice for MX Linux. The main toolbar, which contains the application launcher along with the quick-launch icons and the task switcher, is fairly conventional, but it's located along the left-hand side of the screen to save space. It's a small point, but links to the welcome dialog and the help page on the quick-launch panel are a good idea that will help Linux first timers.

As often happens, the Xfce window manager is set up in a way that means that the active area of window edges and corners are super-small and difficult to drag. Why? Are pixels in such short supply on modern monitors?

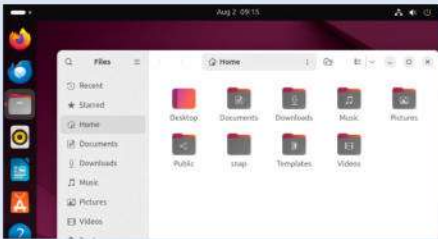
We're not fans of the naming scheme of applications used here. For example, selecting Image Editor actually launches GIMP, but it's difficult to find out which application is installed until you actually run it. It could cause more support problems than it solves.



The Ubuntu desktop layout doesn't vary wildly from that of a stock Gnome 3 desktop. In the Gnome spirit, the desktop layout is simple: a black status bar runs along the top, and the generously proportioned launcher and switcher runs along the left side of the screen. The default Ubuntu colour scheme has become iconic and, generally, everything fits together in a consistent manner.

On the negative side, a lack of detailed customisation refinement might be a frustration for first-timers. Just because they are new to Linux doesn't mean they don't want detailed options for font sizes, for example. Another problem is that the Gnome launcher doesn't handle categories very well, which is a problem if you don't know what you're looking for.

Gnome 3 is simple and consistent, so nearly anyone can pick it up, but switchers might not like the lack of customisation and the stripped-down interfaces.



Visual customisation

Even beginners have ideas about how things should look and work.

There's a smoothness to the way that the Gnome 3 customisation options are operated in Ubuntu. Once you are in the Appearance settings dialog, some of the (clearly worded) options lead to other dialogs. However, by the standards of a typical Linux desktop, the options for customisation are somewhat limited in Ubuntu. In particular, there is no easy way of making detailed font selections or customising the theme.

Pop!_OS, although aimed at power users, doesn't improve on what Ubuntu offers in this department. The Appearance options, for example, simply enable the user to select from the Light or Dark theme. As with Ubuntu, it's possible that these simplified options might be less comprehensive than what switchers are used to. If there is an element that the user can't live with, they might have to dive deep into editing files or adding extensions to change it.

While a little more complicated to customise, Mint's Cinnamon desktop comes with an extensive set of tools for

modifying the look and behaviour of the desktop, making it closer to a traditional Linux desktop in that regard.

Linux Lite relies on Xfce's customisation tools, delivering a good balance between discoverability of settings and full control over how everything looks and works. Of the two, Cinnamon's dialogs are perhaps a bit more polished.

MX Linux has a lot of scope for customisation, but uses a mix of its own and Xfce's tools. Nearly everything can be changed, but it's more complicated to find the right tool to use, and there's a possibility of overlap between different areas of the settings.

VERDICT

MX LINUX	7/10	POP!_OS	6/10
UBUNTU	6/10	LINUX LITE	8/10
LINUX MINT	9/10		

The Gnome-based desktops feel dumbed down, MX Linux feels like it's for experts, while Mint and Linux Lite offer a good balance.

Linux Mint

9/10

The Linux Mint desktop keeps things conventional as it features a layout that will be familiar to most: a combined task switcher/status bar and app launcher sits at the bottom. Having said this, it's a clean and attractive implementation of this approach, with good clarity and contrast. This consistent colour scheme is maintained when system tools such as the file manager are opened, and in our opinion, the fit and finish is as attractive and clear as the Ubuntu desktop.

Without wishing to put down the common combo of Xfce and *Thunar*, this combination of Cinnamon and *Nemo* feels like a quality upgrade, even if it's not quite as lightweight on limited systems. For one thing, the app launcher is bigger and has more information on screen at once.

We feel that a newbie will be impressed with this professional-looking desktop, even though it doesn't break any new ground in terms of layout or functionality.



Pop!_OS

7/10

The Pop!_OS desktop is a heavily customised version of Gnome. With the dock along the bottom, it's a bit closer to the Mac desktop than the Windows one, but it doesn't exactly copy any existing desktop. It's good to see some innovation, but as a result, first-time users have a few things to learn.

If the application launcher is opened in palette mode, the category folders are listed along the bottom, correcting a shortcoming of many Gnome-based desktops. If the search-based launcher is opened using the Super key, pressing the ? key shows the array of search options available from this dialog, and file or web searches or terminal commands can be executed without touching the mouse.

Once you've learned how all this works, you can become a speed demon. Pop!_OS represents the best of what the modern Linux desktop has to offer, but might not be the best starting point for a total newbie.



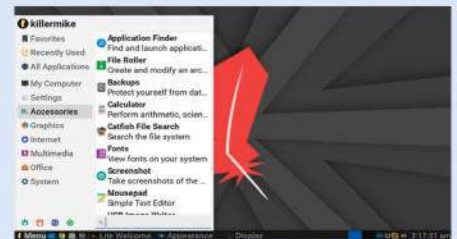
Linux Lite

7/10

As part of its design ethos, Linux Lite makes a conscious effort to replicate the Windows layout. But, in all fairness, a lot of Linux distros use the standard layout of the status bar and launcher running along the bottom. This implementation of that standard scheme, which uses Xfce, employs a fairly monochromatic black/white/grey with occasional use of colour for icons and the odd use of transparency without ever being garish.

Thunar is a fine choice for a file manager as it's extremely light and yet supports network browsing, file previews and multiple tabs.

This colour scheme has a reasonable amount of contrast and is perfectly clear but makes heavy use of midtones, and it doesn't pop out of the screen or have much identity the way that the Mint or Ubuntu desktops do. It looks like a standard business desktop that most people will be able to quickly grasp.



Available documentation

One of the most important considerations for a Linux beginner.

Ubuntu's popularity means that it's the best supported desktop Linux out there. We don't have room to list all the official forms of documentation and tutorials on the website, while tutorials on the web often assume you are using Ubuntu. The official forum is split into different subforums and it's fairly busy, meaning that nearly all questions receive a reply.

Linux Mint is also an extremely popular distribution and benefits from the fact that it has a shared infrastructure with Ubuntu and Debian. Nine times out of ten, you can fix a Linux Mint problem or figure out how to do something by simply following the instructions that would work under Ubuntu. Occasionally, there are differences, but Linux Mint has a big enough community to carry it.

Linux Lite is derived from Ubuntu, meaning a lot of solutions for that distribution work here, too. The site is marred by the prevalence of adverts, but contains a set of well-illustrated articles specifically relating to Linux Lite and a forum that has a healthy level of traffic.

The Pop!_OS website surprised us with how extensive its collection of short articles and videos that focus on the unique aspects of the distribution is. The focus is on areas in which Pop!_OS is different from its parent distribution, Ubuntu, which makes sense.

MX Linux has Mepis, Debian and antiX roots, and it features some good home-grown tutorials and videos on the website, and a forum with a healthy level of traffic. However, tracking down information might be overwhelming for people who are less familiar with Linux.

VERDICT

MX LINUX	6/10	POP!_OS	8/10
UBUNTU	10/10	LINUX LITE	7/10
LINUX MINT	9/10		

All the distros had acceptable support, but Ubuntu leads the pack due to official documentation and third-party tutorials around the web.

Application installation

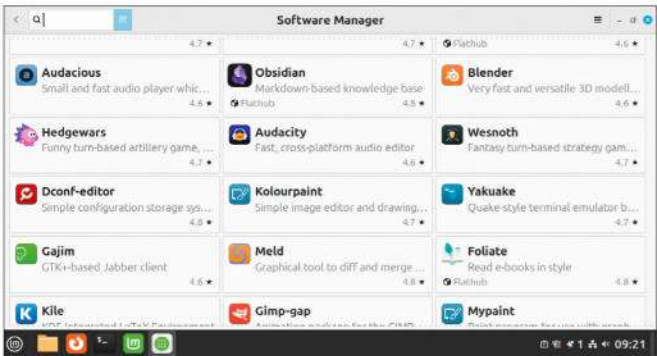
Installing apps is the next step once the basic customisations are done.

Linux Lite has its own software installer that offers a small, curated selection of software. These are the kind of heavy hitters (such as *Audacity*, *OBS Studio* and *Microsoft Edge*) that a convert from Windows is likely to be interested in. 'Package Manager' is actually *Synaptic*, an excellent choice for package control on DEB-based systems. The problem is that if the application the user wants isn't on Linux Lite's custom installer, they have to use *Synaptic* or the command line, which might be too complicated for a newbie.

We preferred the approach employed by Linux Mint because this combines *Synaptic* (for the low-level package management) with *Software Manager*, a GUI with ratings and reviews, and a massive library of applications. This is plugged into Mint's repositories and Flathub.

We'd say much the same for Ubuntu, because it has a high-level package management application, *App Center*, connected both to Ubuntu's repositories and Canonical's own Snap system. What's missing is something like *Synaptic* for the low-level installation of components.

MX Linux has its own application installer, with a huge list of categorised installable applications. It's fairly barebones, and



Linux Mint's Software Manager is a top-class tool for the discovery and installation of apps. As it's connected to Flathub, it's usually possible to grab the latest version.

lacks reviews and ratings, but it does enable you to choose the MX Linux test repository or the Flathub repository in addition to the main MX Linux repository. *Synaptic* is also present.

Pop!_OS's software manager, *PopShop*, is a high-level app store that's plugged into the APT system and the Flathub repository. It's also the system update manager, which makes sense, and this means that you get a summary of updated apps within *PopShop*. It's comparable to what Ubuntu offers.

VERDICT			
MX LINUX	7/10	POP!_OS	8/10
UBUNTU	8/10	LINUX LITE	7/10
LINUX MINT	9/10		

Mint covers both the high-level and low-level installation duties with aplomb. The others tend to be stronger in one area or another.

Default applications

We like a distro that can handle the basic tasks without extra work.

We switched off Linux Lite's generic app naming to get a better overview of the default selections. Significantly, the choices are cross-platform favourites that should be recognisable. The included applications include office tasks thanks to *LibreOffice*, media playback thanks to *VLC*, and image-editing in *GIMP*. There is also a decent selection of smaller utilities for tasks such as file backup and a partition manager. This default selection could cover 90% of everyday tasks.

The Ubuntu selection covers similar ground, but the choices tend more to classic Gnome staples rather than well-known cross-platform apps. Music manager *Rhythmbox* and photo manager *Shotwell* follow the Gnome ethos of simple interfaces with easily discoverable functions. These apps are highly usable for average users, but not very well known outside of Linux.

Linux Mint also covers the basics with *LibreOffice* and a selection of small Gnome-associated utilities, but we wonder if a newcomer might have liked a few extras here and there.

MX Linux has the greatest number of installed applications here. Many of the selections have a gadgety feel and there is a tendency towards lightweight choices. *Strawberry* music



Pop!_OS comes with a set of apps and utilities that cover the basics. In its favour, the selections should present a consistent experience for a user unfamiliar with Linux.

organiser is a fork of the excellent *Clementine* of KDE fame and *Geany* is a fairly comprehensive text editor. MX Linux also has the largest selection of custom utilities. This all adds up to is a mix of apps without much consistency but it does cover most areas.

Pop!_OS comes with a slim collection of mostly Gnome-derived apps. They cover the very basics with consistent-looking apps, and we expect most users to have to add a few more.

VERDICT			
MX LINUX	8/10	POP!_OS	7/10
UBUNTU	8/10	LINUX LITE	9/10
LINUX MINT	7/10		

Ubuntu comes with Gnome staples, while Linux Lite instead favours well-known cross-platform applications.

The verdict

Beginners' distributions

Mint inherits many of the advantages of Ubuntu but places a smart-looking, highly customisable user interface on top. Just because a user is new to Linux, it doesn't mean they won't want to make some customisations. The app choice is basic, but it deviates a bit from the cult of UI simplification that afflicts Ubuntu with choices such as *Nemo* rather than *Gnome Files*, and *xed* is a chunkier option than Gnome's text editor.

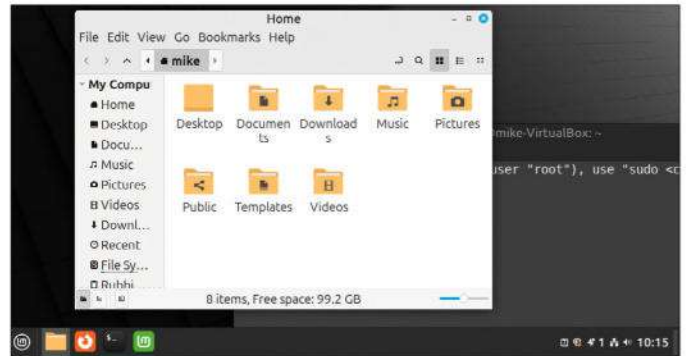
Because of its similarity to Ubuntu and Debian, added to the fact that it's popular in its own right, a user should be well covered in the areas of online documentation and community support. A rolling update system is possibly a better choice for non-veteran Linux users, because it means that full distribution updates are less frequent.

Ubuntu represents a baseline of what a desktop Linux distribution can offer. Its user interface is clear and easy to use and the technical underpinnings are completely sound. Where it falls short for Linux beginners is that the user interface isn't quite like anything else and it's also rather lacking when it comes to customisation. Any customisation beyond the basics is either locked off or only achievable via plugins or editing text files.

Linux Lite presents the familiar desktop layout, and it would make a good office desktop replacement for Microsoft Windows. Underneath it all, it's built on Ubuntu, which means that an administrator who knows Linux should be able to fix any problems that crop up.

We can't rate the Pop!_OS desktop experience as highly as we might like because we're looking at it from the perspective of a beginner. Once you have learned to use this desktop, it's highly efficient and capable of a lot, thanks to everything you can do from the pop-up search dialog. The Ubuntu LTS underpinnings mean that it's well supported and well documented.

MX Linux has a techy feel to it, meaning that it wouldn't be our first choice for non-experts or people who are new to Linux itself. The developers have decided to add applications that draw from a wide selection of toolkits and desktop environments. We admire the fact that MX Linux features a large selection of custom utilities created especially for that distribution, but we worry that this, too, might prove distracting to new Linux users.



1st **Linux Mint** **9/10**

Web: <https://linuxmint.com> **Licence:** Various mostly open source licences **Version:** Linux Mint 22 Cinnamon edition
Great-looking desktop. Solid Ubuntu underpinnings. Good software centre.

2nd **Ubuntu** **8/10**

Web: <https://ubuntu.com> **Licence:** Various mostly open source licences **Version:** 24.04 LTS
Simple, easy-to-use desktop. Widely supported. Limited customisation.

3rd **Linux Lite** **8/10**

Web: www.linuxliteos.com
Licence: Various mostly open source licences **Version:** 7.0
Basic implementation of a standard desktop. Cross-platform applications.

4th **Pop!_OS** **7/10**

Web: <https://pop.system76.com> **Licence:** Various mostly open source licences **Version:** 22.04 LTS (42)
Slick, powerful interface with useful features. Aimed at power users.

5th **MX Linux** **6/10**

Web: <https://mxlinux.org> **Licence:** Various mostly open source licences **Version:** 23.3
Many custom features and utilities. Complicated support.

» ALSO CONSIDER

Zorin is a distro with slick desktop designed with OS switchers in mind. It has a Pro paid tier as well, which can sometimes be a good idea as it means that there is a funding stream.

Elementary OS is another distribution with a highly polished desktop. Like Zorin, its creators have specifically considered the needs of switchers. It's donationware, but you can select an amount of £0 before downloading if you want to see it in action before donating.

Deepin Linux has a nice, custom desktop and it's supposed to be easy to use. We're sure the rumours that data is being sent to Chinese servers are just that, rumours.

Another approach that is worth thinking about is to build a desktop specifically for the user who will be using it. Doing this, you can customise things to look like the desktop that the user is used to. This means that you've got the choice of all of the best tools. **LXF**

BRIGHTER SMARTER HOMES!

Matt Holder **uses Home Assistant and Zigbee devices to make automations and dashboards showing the state of our homes.**

Over the last 10 years, home automation has really taken hold and many companies have wanted a piece of the action. Google, Samsung and Apple all have their own software and hardware ecosystems that can discover, group and integrate smart home gear, such as smart speakers and other tech. It's certainly convenient, but it's also locked in and monitored.

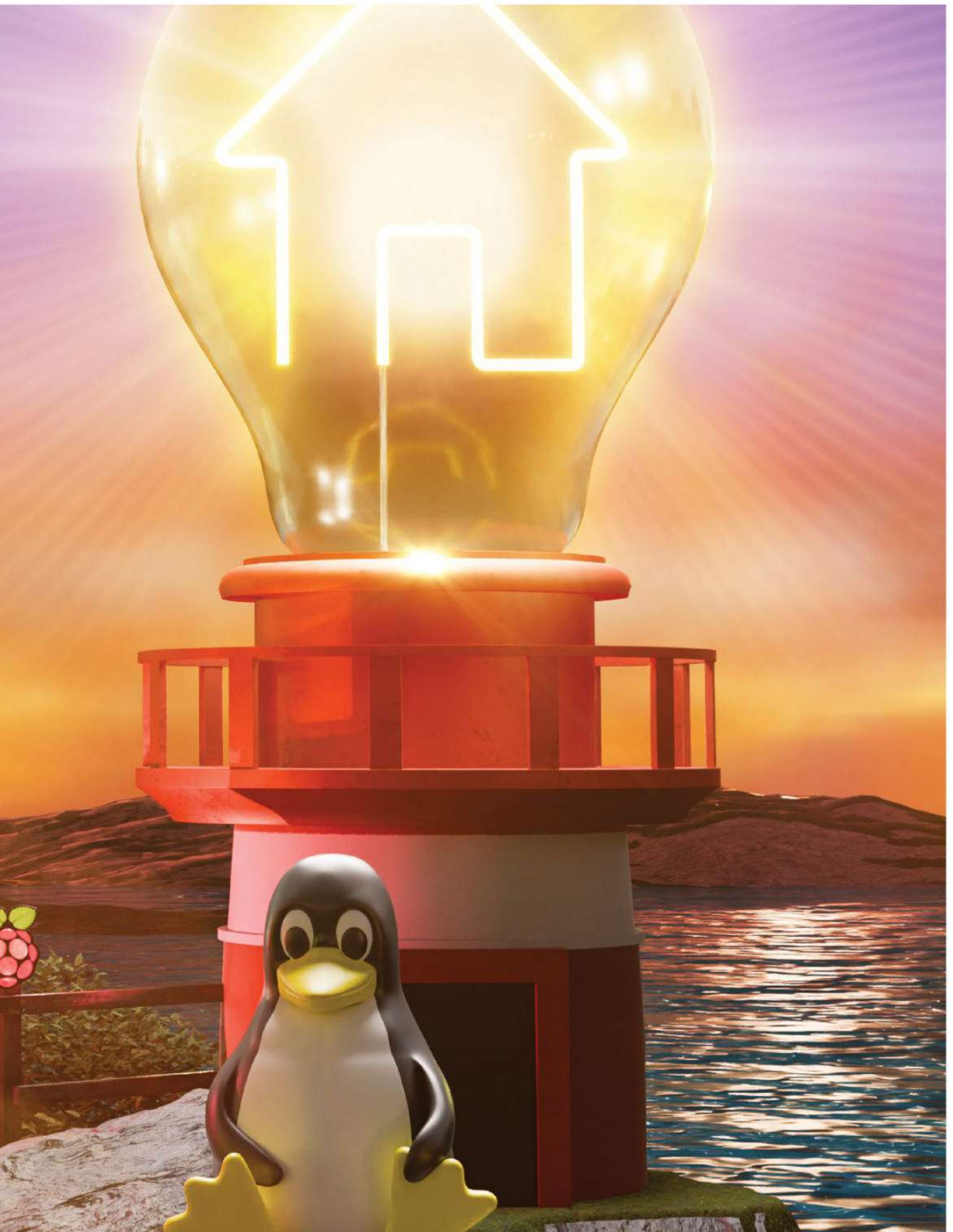
So, what are we going to do about it? Firstly, we are going to install *Home*

Assistant on a Raspberry Pi (a spare PC or home server will do, too) and have a quick walkthrough of the GUI. *Home Assistant* is an enormous project and we can only scratch its surface, so we're focusing on connecting and controlling standard Zigbee smart lighting and creating some useful automations to make our lives easier.

Along the way, we'll talk about some of the protocols (such as Zigbee) that are available, any hardware needed and where we can get suitable devices to use. We will then move

on to the integration of these smart home devices for easy control.

Home Assistant (HA) is an open source project, started by Paulus Schoutsen, with the goal of integrating as many different data sources as possible, while keeping as much of it as possible local to our networks. Its ownership has been transferred to the Open Home Foundation, which also owns other projects, such as *ESPHome*, and aims to be a moral steward and stop any projects going down the wrong track. So, let's start making our homes a little smarter and brighter...



Making you smarter



Before kick-starting your new obsession, it is worth explaining a little bit about Home Assistant's concepts...

QUICK TIP

Instructions for how to install Home Assistant can be found here: www.home-assistant.io/installation. For a dedicated PC installation: <https://bit.ly/lxf320pc>. To install and run it from within a VM: <https://bit.ly/lxf320vm>.

The Home Assistant operating system ties a number of different things together: the OS itself, Supervisor and Home Assistant. Supervisor provides an environment to manage the OS, provide updates and also run containers, which can interact with Home Assistant. Once familiar with the software, it is highly likely that it will become a permanent part of your network. Follow the guide below for a Pi install or see links to the left for PC and virtual machine installations.

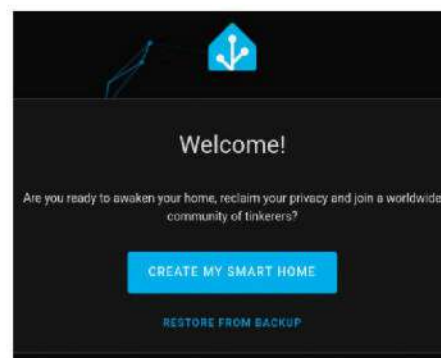
Given this, it is a good idea to set a static IP address. This can be achieved by using the Supervisor options. On the left-hand side menu, go to Settings > System > Network. Use the choices here to change the relevant options. Once completed, wait a minute or so and then connect to the web interface once again, but this time using the new details.

Navigating back to Settings, you now have access to the Add-Ons tab. Within this, you will see an overview of any installed add-ons. These can be installed from the Add-On Store link and cover a

wide range of things, such as AdGuard Home DNS filtering, DHCP servers, MySQL database servers and NodeRed, which can be used to interact with data from Home Assistant.

As will be discussed shortly, extra functionality can be added to Home Assistant using integrations as well as add-ons. Add-ons are Docker containers that the Supervisor looks after and can be provided by the Home Assistant core team or any member of the public. Integrations are more tightly linked to Home Assistant and are provided as part of the core release. The Settings > System > Backups options allow snapshots to be taken, to which you can roll back should upgrades or certain configuration changes cause the system to break. Finally, the Settings > System > Hostname option enables the Hostname to be changed. Using the charts in Settings > System > Hardware, an overview can be seen of RAM and CPU usage. Disk usage can be seen in Settings > System > Storage, while Settings > System > Logs provides a view of the logs, for troubleshooting purposes.

INSTALL HOME ASSISTANT OS



1 Download the Raspberry Pi Imager

The Raspberry Pi Imager tool was created by the Raspberry Pi Foundation to make writing OS images to microSD cards as simple as possible. Navigate to www.raspberrypi.com/software/, then download and install the software. Once opened, select the Device option and select your device. From the Operating System section, find the Home Assistant option and then from the Storage section select your memory card. Use the Next option and any further dialogs to start writing the image.

2 Prepare your Raspberry Pi and turn it on

Once your microSD has been written, eject it from the device writing the image and connect it to your Raspberry Pi instead. Next, connect your Pi to power and use a network cable to connect it to your router. This can be temporary and if you want to use Wi-Fi instead of a wired connection, this can be configured later in the process. Once booted, you will see a screen like the one above, which details how to access your new installation.

3 Start the onboarding experience

Once booted, open your browser and enter <http://homeassistant.local:8123>. Once loaded, click on the Create My Smart Home button and follow the wizard through. Steps include creating your first user account, setting your location and choosing what you are comfortable sharing with the project – any data that is shared is anonymised, so shouldn't cause too much concern.

We will now have a look at some of the other configuration options available. Within the Settings menu, a lot of things can be changed. For instance, Settings > Devices & Services is where the details are stored about the integrations that were discovered during the onboarding experience (and where items added in the future can be found as well). Within the Integrations tab, a large number of other integrations can be set up. Have a look at what is on offer to see what can be added to your own installation. The Devices tab can be used to see the items on our networks that have been configured, Entities shows all entities registered on the system, while Helpers enables virtual entities to be added – such as grouping switches or lights, creating controls and creating entities, which can average values from a number of others or sum values over time.

Go to the Settings> Areas, Labels & Zones > Areas options to be able to group devices and entities in different rooms within your house. Labels is a relatively new feature, which enables labels to be applied to different object types. The Zones tab can be used to create geographical areas that can be used in automations. Dashboards are used to create overview pages, where many types of data can be displayed and these can be interacted with from Settings > Dashboards. Navigating to Settings > People provides options to create people from within your household and control whether they need to be able to log in or not.

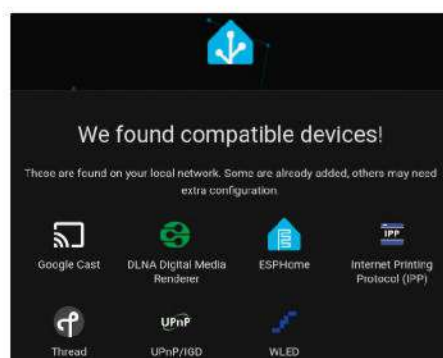
On the left-hand menu, the Logbook and History sections can be used to view changes to your entities



over time, while the Overview option is used to view the default dashboard.

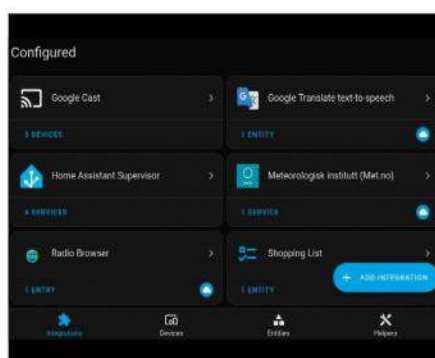
Another useful section to introduce is the Developer Tools page. This is available from the sidebar. Developer Tools may need to be switched on from the User settings page (accessed by selecting your name at the bottom of the menu and turning on Advanced Mode). The States tab within Developer Tools allows values to be seen for all entities. This screen provides a quick and easy way to see when values change. The screen can become confusing, though, when there's a large number of entries present.

Here's an example of what can be achieved using the Home Assistant interface.



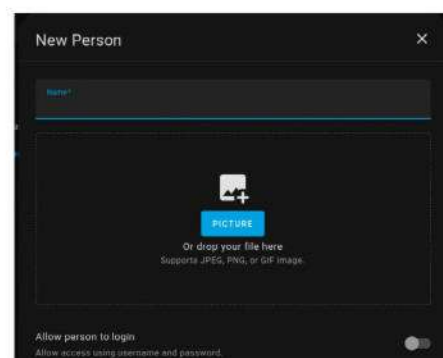
4 Perform device discovery

The final step of the onboarding wizard is to perform a scan of your network and look for supported devices that can be interacted with. This section is purely informational and doesn't allow direct configuration. Spend some time researching what has been found. For example, Google Cast is especially useful because it can be used to play local audio files and play announcements, using speech-to-text, such as informing whoever is currently setting the thermostat too high!



5 Investigate available integrations

Once the wizard has been completed, use the Settings > Devices & Services options to configure any integrations that you want to use. Configured integrations are already displayed and the + Add Integration button can be used to investigate other integrations and add them to your Home Assistant installation. You can visit <https://analytics.home-assistant.io/integrations/> to see what other Home Assistant users have installed on their systems.



6 Add your household

Navigate to the Settings > People > Add Person button to add all people from your home. The People features allow various types of functionality to be set up, such as noting the presence of people when they are at home or away. Registered people can also be allowed to log in to the installation, if this is needed. Dashboards can be created for specific users and/or purposes, which would require your family members to log in to the system.

Create dashboards



To control your smart home, you need dashboard controls that link to your smart home devices.

The system within *Home Assistant* that can be used to build dashboards is called Lovelace.

Multiple dashboards are supported and they can all be designed in different ways. This is incredibly useful, because it means that layouts can be designed for different screen sizes. For example, you could have one dashboard for a smartphone and another for a tablet. By default, *Home Assistant* builds a dashboard that contains all the different items that have been added to it.

To create a new dashboard, head to Settings > Dashboards and then click on the Add Dashboard button. Four types are supported and the one we are creating requires us to select the New Dashboard

selecting the pencil icon in the top-right corner. At this point, you will see a completely empty dashboard, which is in edit mode.

Dashboards can contain multiple tabs or views, and each tab can have its own name or icon, which can be used to select it. Tabs can also be set to have different modes. To edit a tab's options, click on the pencil next to its name/icon. The view type options are Masonry, Sidebar or Panel – Masonry refers to a system in which multiple cards are spread across the screen, Sidebar features two columns, one wider than the other, and Panel mode allows just one card to be shown on the screen, which is useful for something like a kiosk or tablet device. Within the tab's options, the

Badges section can be used to add small icons at the top of the screen to display various values, such as temperature or the status of a switch.

The Background options allow a background image to be set, and finally the Visibility section allows tabs to be hidden for certain users – maybe only power users should see the battery status of remote controls, for instance, or perhaps

you would rather not allow your children to be able to switch lights on and off. Change the settings as required and then select Save.

We've omitted to mention Sections – these are currently experimental and allow for the dragging and dropping of entities into different columns on the screen. This feature shows lots of promise and is likely to be the future of *Home Assistant* dashboards. However, due to its experimental status, things could change, so you probably shouldn't rely on this just yet.

DASH IT ALL!

“Multiple dashboards are supported and they can be designed in different ways, so you could have one dashboard for a phone and another for a tablet.”

From Scratch option. Once a title has been added, an icon selected from the icon picker and other options chosen, click on Create. Next to the dashboard that has just been created, select the arrow icon or Open button, which opens the new dashboard. This dashboard contains nothing at this point and can be edited by

» OBJECT TYPES

Home Assistant is an incredibly powerful tool and has lots of different concepts to learn about. Devices represent physical items that have been configured on the network – for example, a heating thermostat or a Chromecast device. Entities represent the capabilities of these devices. For example, a heating thermostat has entities referring to whether the heating is currently switched on or off, what the target temperature is and what the current temperature is. A Chromecast device has an entity associated with it that provides media playback. Services are used to send signals to devices or entities to make changes. For example, using the correct service, a text-to-speech message can be sent to a Chromecast device or the heating could be turned on.

Entities have types and are named as such: TYPE.NAME. A few examples of these types are `binary_sensor`, which represents a true/false value; `sensor`, which represents a numerical value; `switch`, which represents something that can be switched on and off; and `media_player`, which provides media playback capabilities.

Display cards

The next step in adding items to the dashboard is to add what are referred to as cards. These cards can contain a single piece of information or multiple entries. If added individually, Lovelace automatically determines where to place them. To have further control on where the cards are placed, horizontal stack and vertical stack cards exist, which enable cards to be placed next to each other in the defined order. There is also a grid card, which can be used to create a table-style view of cards.

Click on the Add Card button to allow the dashboard to be designed. The designer is split into two tabs: By Card and By Entity. By Card is used to find the sort of card you'd like to display, then when it has been selected, the entity to be displayed can be added. When using the By Entity section, you first search for

an entity and then suitable cards are displayed. Both do the same thing, but each accomplishes the task from a different angle.

The first thing we will add to the dashboard is the position of the sun. With the By Entity tab selected, enter 'sun' into the search box. When Sun is displayed, select the checkbox next to it and click on Continue followed by Add To Dashboard. The second thing we will add is a series of buttons, which we can use to quickly and easily visit another website and the configuration page of *Home Assistant*. Click on the Add Card button and this time select the Horizontal Stack. Next, select the Button Card. On the options that appear, remove any Entity value that is auto-populated and change the Tap Action to URL. For Name, enter 'Linux Format' and in the URL Action box, enter http://www.linuxformat.co.uk. Note the Hold Action option, which can be used to set a second action if the button is pressed down for a longer period of time. When the options have been selected, click on Save.

Now edit the last card and add another button entry using the plus option. This time, set the Name field to Configuration, Set Action to Navigate, and Navigation Path to /config. Finally, save the options for the current card and then exit the dashboard's edit mode by using the cross in the top-right corner.

To experiment with the different view modes, edit the tab's settings and change the view mode to whichever suits you. Now this has been completed, your dashboard contains a card displaying the position of the sun and a couple of buttons that can be used to accomplish useful tasks (see screenshot, below-right). Buttons can also be used to trigger services and toggle switches. Cards also exist to graph numerical information over time, view live camera feeds, display floor plans with icons overlaid, as well as many other things.

Integration technologies

The main two consumer-grade integration standards that you can buy kit for are Zigbee and Z-Wave. Both of these technologies are well known, but Z-Wave tends to be more expensive. Anecdotal evidence points to Z-Wave being more reliable, but our experience of Zigbee has been relatively positive. Both Zigbee and Z-Wave devices can be mains or battery-powered, and are optimised for low power usage and long battery life. We will focus on Zigbee, which we have experience with.

Zigbee operates on the same frequency as your base Wi-Fi – 2.4GHz. This can cause issues with cross-talk and interference, and it is generally recommended to connect your dongle to your Raspberry Pi via a USB extension cable. Zigbee builds a network starting with the controller, then devices connect to this. Most mains-powered devices, such as bulbs and smart sockets, are able to act as repeaters, which means that the network is strengthened by devices

» BACK UP YOUR SYSTEM

A key to making sure your home automation setups are as reliable as possible is to take regular backups. New installations of *Home Assistant* provide an option to restore from backup, so should anything go wrong, data recovery is very simple. Creating backups is made really straightforward from *Home Assistant* itself and is even easier using an installable add-on. To create a one-off backup of *Home Assistant*, navigate to Settings > System > Backups. When this has been selected, you will see a button to create a backup, and you can interact with existing backups. A brilliant add-on called *Home Assistant Google Drive Backup* is available from the add-on store that creates and manages backups that are then stored in Google Drive. Installation instructions are available at <https://github.com/sabeechen/hassio-google-drive-backup>. Once installed, options are available to run at startup, enable the watchdog and auto-update. Select the add-on and follow the installation wizard to authorise access to your Google account. Once authorised, investigate the other options, which enable you to select the backup frequency, timings and number of backups to be kept.

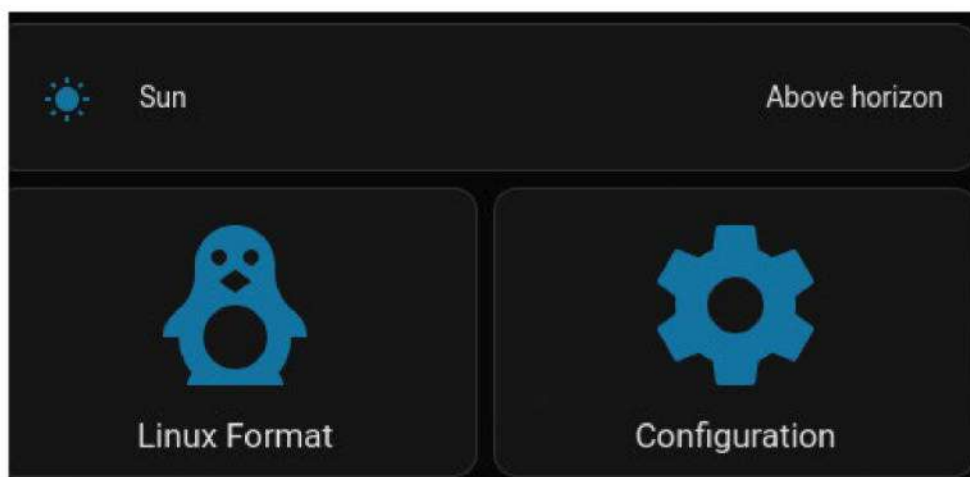
connecting to a nearby neighbour, before ultimately connecting back to the controller. Zigbee networks are also self-healing, meaning that if the nearest repeater drops offline, devices connect to the next nearest one, or the controller.

POWER POINTS

“Both Zigbee and Z-Wave devices can be mains or battery-powered, and are optimised for low power usage and long battery life.”

Home Assistant OS uses an integration called ZHA to connect to your USB Zigbee controller, and provides an integration layer to allow seamless interaction with your Zigbee devices. An alternative to the ZHA integration is *Zigbee2MQTT*, which can be used either as an add-on on your *Home Assistant* OS installation, or installed as a container elsewhere and connected back to *Home Assistant* via MQTT.

Building your first dashboard.



Pair your Zigbee gear

Using low-cost widely available smart lighting is easier than you might think, thanks to open protocols and open standards.



QUICK TIP

The Zigbee compatibility repo is an excellent resource when deciding how to configure devices or which devices to purchase: <https://Zigbee.blakadder.com>.

When it comes to automating your home, there's a wide array of products available on the market that make controlling everything a breeze. This is all down to low-cost open standards that can be easily integrated into systems you control. The most widely used are Zigbee, Z-Wave and Thread. These allow instructions, such as a smart light bulb illuminating a certain colour at a specific time, to be sent to multiple devices at once, providing you have a compatible smart home hub that can talk to all of your smart home devices.

Unlike Wi-Fi, these smart home standards use a tiny amount of power, which means many smart home devices can run on batteries that don't need changing or recharging for months or even years.

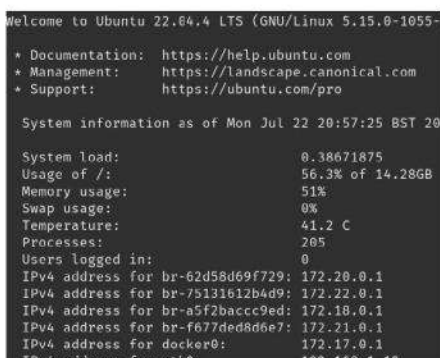
The one we're interested in is Zigbee, which is a wireless networking standard, and its specifications are maintained and updated by the non-profit Zigbee Alliance organisation, which was established in 2002. Zigbee is certainly an unusual moniker – it turns out that it got its name from how honeybees communicate



1 Add and configure your ZHA integration.

with each other by moving, which is also known as the waggle dance. There are more than 400 different technology firms that support the standard, including behemoths such as Apple, Amazon and Google, alongside well-known names such as Belkin,

CREATING A SECONDARY ZIGBEE NETWORK



1 Prepare a memory card

If you need to extend your Zigbee control this adds a second Pi running Zigbee2MQTT. Using *Raspberry Pi Imager* select the Device option and then select your device. From the OS section, find the Ubuntu option, and then from the Storage option, select your memory card. Next, use the Ctrl+Shift+X key combination to open the advanced options and configure any options you want, including account details, language and Wi-Fi connection. When configuring these options, make sure that SSH is enabled if you want remote access.

2 Boot the Pi and login via SSH

If you are not going to be using SSH to connect to your Pi, connect HDMI, USB and power (don't forget to connect the network cable if you are not using Wi-Fi). Log in using your keyboard and mouse, or use SSH from another device to connect. The following command can be used to connect: `ssh pi@IP_ADDRESS`. You may need to log in to your router to find the IP address that has been assigned to your Pi, else check your router or use a WiFi scanner to look for the Pi on your network.

3 Install Docker

We will be running Zigbee2MQTT as a Docker container, so we can separate it from the installed packages on the underlying OS. To install Docker, follow the instructions here: <https://docs.docker.com/engine/install/ubuntu/>. Zigbee2MQTT needs its own configuration and folder structure. Create the directories we need with: `sudo mkdir -P /docker/Zigbee2MQTT/data`.

Huawei, Ikea, Intel, Qualcomm and Signify, as well as many others.

The Zigbee standard can broadcast data across distances of approximately 75-100 metres indoors or around 300 metres in the open air, meaning it can easily offer strong, stable coverage in large homes.

Zigbee operates as a mesh network, which means instructions can be sent between devices attached to the same Zigbee network. In theory, each device acts as a node, receiving and transmitting data to every other device, helping the data spread further, so the smart home network can cover a large area.

However, with Wi-Fi, the signal decreases over distance or can be blocked entirely by the thick walls of an older property, which means the command may not reach the smart home devices that are furthest away, so keep this in mind.

Nice pair

Before pairing any kit, we need to set up *Home Assistant* to be able to connect to it. Unpack your Zigbee dongle (we're using the SkyConnect device) and plug it into the bundled USB extension cable. Now connect the other end to your Raspberry Pi that is running *Home Assistant* (the steps are very similar with other dongles). Open your browser, navigate to http://<your_IP_address>:8123 and log in with the account you created while completing the on-boarding wizard. Using the menu on the left-hand side, navigate to Settings > Devices And Services, then select the Add Integration button. On the menu that opens, search for ZHA, select it and pick the option that

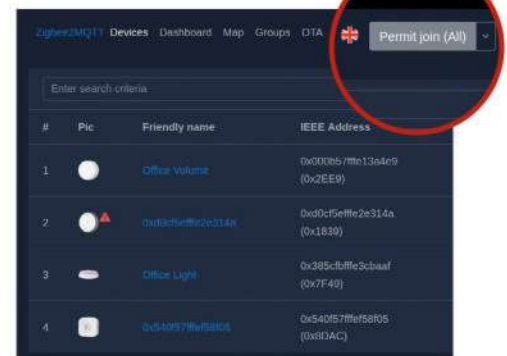
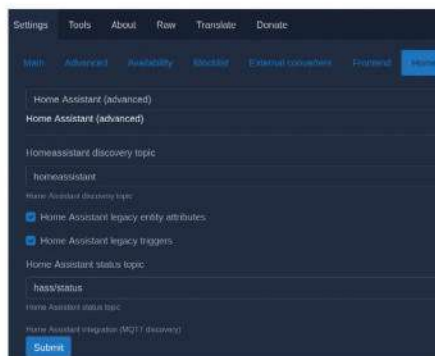
» SMARTPHONE APPS

Interacting with *Home Assistant* via a web browser is a very pleasant experience, whether using a phone or larger screened device, such as a laptop. The *Home Assistant* team and community also make apps available for iOS devices as well as Android. These integrate very well with the phone's operating system and allow the phone's sensors to be provided to *Home Assistant* and displayed in the Lovelace interface. It is also possible to design a bedtime routine in which a *Home Assistant* automation switches off your lights when your phone is charging and on silent. When a smartphone app has been registered with your *Home Assistant* installation, it can use the phone's notification system. For example, if the doorbell rings, a *Home Assistant* registered camera can take a photograph, which can then be sent via notification to your phone. Also, the capability within notifications for the user to be asked a question can be harnessed, so the user can easily send a response back to *Home Assistant*, which can then be acted upon. The momentum behind the apps is high and features, such as Google's Android Auto, are now supported, as well as watch support for various devices. Apps can be downloaded from the Play Store or App Store, and once opened, a wizard can be followed to connect to your *Home Assistant* server.

matches your controller type. The SkyConnect device uses a Silicon Labs chip, so select the top option.

Click Submit and on the next screen enter the serial port used by the device, baud rate of 115200, and select Hardware Flow Control. To set the serial port, you can use `/dev/ttyUSB0`, but it is better to go to Settings > System > Hardware > All Hardware. On this

```
version: '3.8'
services:
  zigbee2mqtt:
    container_name: zigbee2mqtt
    image: koenkk/zigbee2mqtt
    restart: unless-stopped
    volumes:
      - ./data:/app/data
      - /run/udev:/run/udev:ro
    ports:
      # Frontend port
      - 8080:8080
    environment:
      - TZ=Europe/Berlin
    devices:
      # Make sure this matched your adapter location
      - /dev/serial/by-id/
        usb-Texas_Instruments_TI_CC2531_USB_CDC___0X001
        ttyACM0
```



4 Your docker-compose file

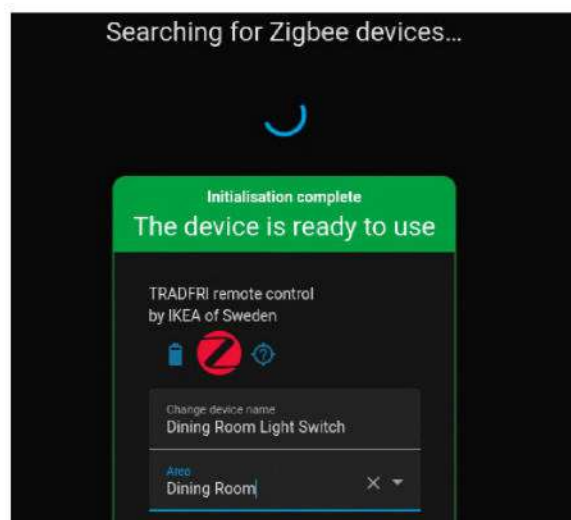
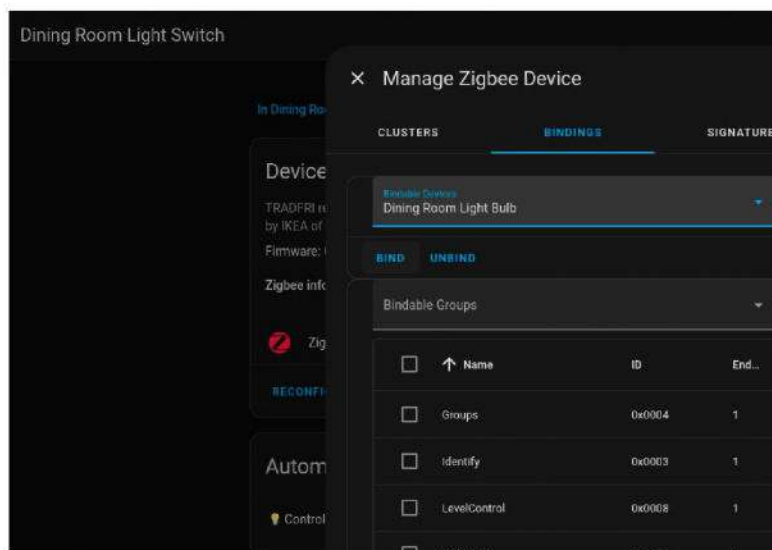
On your *Zigbee2MQTT* Raspberry Pi, change directory to the `/docker/` *Zigbee2MQTT* directory with `cd /docker/Zigbee2MQTT` and open the *Nano* text editor to enter the text above (also here <https://bit.ly/lxf320mqtt>) tweak anything required, such as the time zone and device path for your Zigbee adaptor: `sudo nano /docker/Zigbee2MQTT/docker-compose.yaml`. Save the file by pressing `Ctrl+O`, `Enter`, `Ctrl+X` and then run the container by entering `sudo docker compose up &`.

5 Connect Zigbee2MQTT to Home Assistant

Once your container has started, you will be able to visit the interface by opening your browser and entering `https://IP_ADDRESS:8080`. Click on Cog > Settings > Home Assistant Integration and make sure the options look like those shown above. Also from the Settings menu, select the MQTT tab and enter the Base Topic that will be used (something representative of the Raspberry Pi's location or purpose), MQTT server, username and password.

6 Adding devices

To add a device, make sure the drop-down option at the top-right corner of the page shows Permit Join (All), reset your device and switch on the pairing mode. After a minute or so, you should see your new device appearing on the main part of the page. Use the friendly name link to interact with your device and also rename it. When doing this, ensure that the Update Home Assistant Entity ID button is selected so that it can easily be referred to from *Home Assistant* itself.



Pairing Zigbee devices to your Home Assistant installation.

Binding devices is a very useful technique, especially when an advanced automation is not required.

page, you can find your device and copy the full path, which is useful as **ttUSB0** can change when other devices are connected. Go back to the wizard and once the values have been added, hit Submit.

Now we need to pair your device to the controller. Navigate to Settings > Devices And Services, scroll down to Zigbee Home Automation and click on the text below the icon, which takes you to an empty Devices page. From here, click on + Add Device and a new page opens. At this point, make sure the device you want to pair is ready to go. For battery-powered devices, ensure they are powered up and perform the manufacturer-specific magic incantation to enable pairing. If you are using Ikea Tradfri switches, pressing the button next to the battery compartment four times resets the device and enters pairing mode. For Tradfri bulbs, switch them on, then start the reset procedure, which is to turn the bulb off and on again six times in a row. When the bulb switches on, it pulses and cycles brightness from lowest to highest to show it has reset.

Once the device has entered pairing mode and has been added to Zigbee Home Automation, you'll see a

window on the screen, asking you to name the device and add it to an area. Once this has been completed, you can use your device from automations or dashboards. Repeat the procedure for other devices.

Before creating automations, as a brief aside, a useful part of the Zigbee spec is binding. This means devices can be bound to others and communication can take place directly between them, rather than needing to communicate with the controller and *Home Assistant* then taking an action via an automation. This is especially useful should anything happen to your Zigbee controller or *Home Assistant* installation. Devices can be bound by finding them from the Zigbee Home Automation > Devices screen, selecting your remote or switch, selecting the three dots menu, clicking Manage Zigbee Device, navigating to the Bindings tab, selecting from the Bindable Devices drop-down option and selecting the Bind button.

Introducing automations

Over the last few years, a huge amount of work has been done to ensure that *Home Assistant* is as user-friendly as possible to newcomers. This includes a wizard to support the creation of new automations and the ability to create automations directly from a device's page, with examples of what can be created, which is generated by using the device's capabilities.

Before we create our new functionality, we need to introduce a few simple ideas. The first is triggers – these are what are used to fire an automation. If multiple triggers are added, they operate using OR logic. The next is conditions – once an automation has been triggered, the conditions are evaluated to determine whether the actions should be fired. If multiple conditions are added, the default is that they operate using AND logic (the opposite of triggers). Conditions can be grouped into blocks, with AND and OR operations together to create very complex conditions to be met.

Finally we have the actions, which are the output of the automations. Many building blocks can be used, so that actions can be conditional, run as a sequence or in parallel, and can even operate depending on which trigger caused the automation to be fired. We describe a lot of this functionality on the final page.

» FRIENDLY MOSQUITOES

MQTT stands for Message Queuing Telemetry Transport and is a messaging service that is accessible by a wide range of hardware. Libraries exist for devices all the way from x86 to microcontroller boards. The messaging protocol requires very little processing power and a tiny amount of bandwidth. MQTT operates as a publish-subscribe system – an MQTT broker runs somewhere on the network, which devices can publish to; other devices then subscribe to the broker and are able to act when certain messages are received. The recommended broker for *Home Assistant* is *Mosquito*, which can be installed from the add-on store (Settings > Add-Ons). Once installed, navigate to the Users section of the configuration and add a new user for MQTT. This can be a limited user – it doesn't need to be an administrator – and can be set to allow access from the internal network only. Now that the MQTT add-on has been installed, head to the Integrations page within Configuration and configure the MQTT integration that's been discovered. The *Home Assistant* team has developed a standard so that devices can be automatically added when certain messages are received.

Automatic pilot

Automations are key to creating a smarter home and ensuring it works from dusk to dawn.



The automation that we will work on first is to switch on the lights after sunset and before sunrise when entering the room. *Home Assistant* provides a sensor, which tracks the location of the sun, and we can pair this with our newly integrated smart bulb. This example also uses an Ikea Tradfri motion sensor. Navigate back to *Home Assistant* and from the menu on the left, navigate to Settings > Automations & Scenes. Use the + Create Automation button and select the Create New Automation option to create a blank one. Now, in the Triggers section, use the Device option, and in the Device box, search for your motion sensor. In the Trigger section, select the option that resembles motion being detected. We now need to add a condition, so that the light only turns on if the sun has set on an evening and not risen again the following day. To do this, use the + Add Condition option and search for Sun and select Before Sunrise and After Sunset. Offsets can also be added if the time needs to be shifted slightly. We now need to add our actions, so under the Then Do section, use the + Add Action to add the action, which turns a light on. Select Device and search for the light you want to turn on. Under the Action section, select the Turn On option. If your bulb supports it, you can also set a brightness at this stage. At this point, click on Save and name the automation – please note, emojis can be used within automation names, which is an easy way to categorise your automations. The Category functionality could also be used (see the Quick Tip).

This automation should now work, but doesn't have any functionality to turn the light off again after the motion ends – we can now add this. Edit your automation and navigate down to the Actions section. Add another action and use the Building Blocks option to find the more complicated functionality. Select the Wait For A Trigger option and select + Add Trigger, select the Device option again, and search for your motion sensor again. Under the Triggers option, select the one that relates to motion not being detected and add a duration of 30s. What this does is it pauses the automation after switching the light on and waits for the motion sensor to stop detecting motion for a 30-second period. Once this has occurred, the next stage runs. Add

another action to add your light again and this time select the option to switch it off.

To be able to organise automations further, it is possible to have multiple related flows in one. To do this, when adding a trigger, select the ... icon and click on Edit ID. Add some text to the Trigger ID field that is now shown. When adding actions, you can now use the building blocks to add the Choose option and from within this, add a condition and search for Triggered By. Once this has been added, you can select the trigger ID that you want to trigger this part of the actions and then add any actions. You can then add as many options as you like to the Choose block, allowing many different automations to be created in one place. This is very useful, but can also make troubleshooting more difficult, so use wisely.

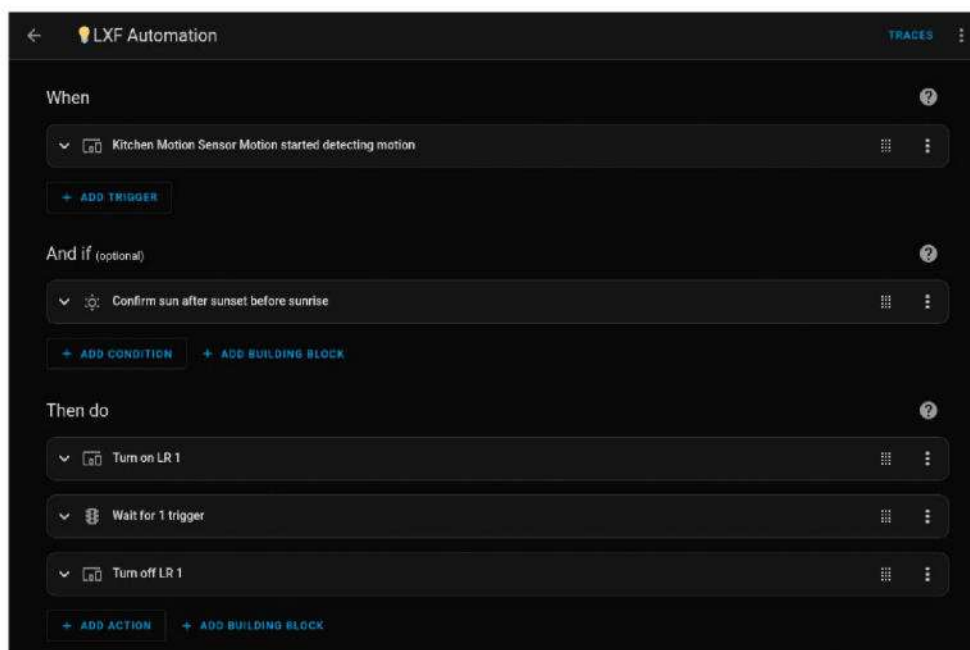
Before bringing the article to a close, we need to talk about *Zigbee2MQTT* (see *walkthrough on page 38*). As previously mentioned, this can be used as an alternative to ZHA or as a way of providing a second network elsewhere in your house by using the MQTT functionality. The integration is seamless and you would think your device is connected directly to the Zigbee coordinator that is connected directly to the Pi that's running your *Home Assistant* installation.

We trust that you have enjoyed this whistle-stop tour of *Home Assistant* and it has given you some ideas of what you can achieve in your own home. Privacy is really important and big companies knowing information about your home is often not a particularly positive thing. **LXF**

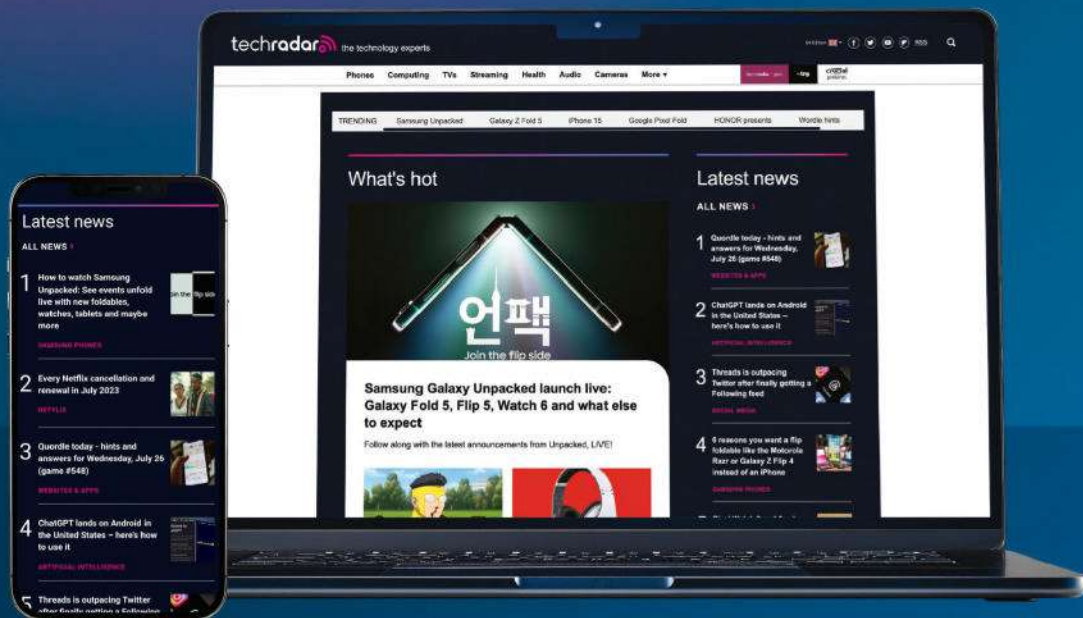
QUICK TIP

Categories can be used to organise items within *Home Assistant*: www.home-assistant.io/docs/organizing/categories/.

Our first automation turns a light on and off, based on motion.



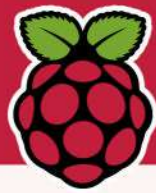
Meet the technology experts



- The world's **most** comprehensive technology website
- An **unrivalled** mix of news, opinions, reviews and features
- **All-new design**, new homepage, new features and special reports
- Backed by **over 300 years** of editorial experience

techradar
the technology experts

www.techradar.com



Shoot for the Moon with Moonhack 2024

Join in with the Australia-based hackfest that's open to everyone around the world!

Formed in 2017, Moonhack is an Australian-run hackfest and event created to help encourage children to code and work together on physical projects. While it's organised by Code Club Australia, it's open to children from anywhere in the world.

Aimed at children aged from eight to 15, Moonhack 2024 opened for registrations on 1st September 2024, while the event runs from 14th October for two weeks. You can submit projects until 30th November and winners will be announced on 8th December.

The Moonhack events enable participants to

choose one of six projects to tackle. They range in difficulty, so it's easy to choose one for your level. So what are you waiting for? Register today and have some coding fun! Find out more at <https://moonhack.com>.



Enhance your coding and making skills!



Les Pounder works with groups such as the Raspberry Pi Foundation to help boost people's maker skills.

» GETTING INTO INTEL

At Tom's Hardware, I've recently been testing single-board computers (SBCs) that use Intel CPUs rather than the Arm processors found in the Raspberry Pi. The Intel N100, an Alder Lake N-based CPU, has four cores, four threads and turbos to 3.4GHz. Sure, it uses a little more power – 6W versus the Raspberry Pi 5's 2.7W at idle – but the inclusion of an Intel CPU means that essentially we have a tiny desktop PC that can run any of the Linux distributions that we love. And yes, it can also run Windows 11, should you really want to.

The Radxa X4 (<https://radxa.com/products/x/x4/>) is the latest such SBC to cross my desk. It apes the Raspberry Pi form factor and provides pretty decent performance for less money than the Raspberry Pi 5.

Other SBCs often fall short of the Raspberry Pi in one key area: the GPIO. However, the Radxa X4 has a trick up its sleeve: it has a Raspberry Pi RP2040 on board. Yes, the same chip as the Raspberry Pi Pico, and the 40-pin GPIO mimics the Raspberry Pi pinout but provides all of the features found in the Pico. We just need to do a little Linux magic to communicate to the Pico via serial using *Thonny*, and we are good to go.

At the moment, it is still early days for testing, but so far I am pretty impressed with the mix of Intel performance and Raspberry Pi GPIO. Keep an eye on www.tomshardware.com/author/les-pounder for the full review.

The Pi 5 GPU Hacked, of course!

Pineboards, a maker of Raspberry Pi expansion boards, has shown off *SuperTuxKart* running at 4K on a Raspberry Pi 5 via an external PCIe AMD Radeon RX 460 graphics card. With a bit of driver hacking and an expansion card, they got everything up and running. Find out more here: <https://bit.ly/lxf320pigpu>



Multiple times the size and power of a Pi.

Pi 5 2GB New low cost.

For £47, you can get your hands on a Pi 5 but with a reduced 2GB of memory. The Pi Foundation held back on this model as its hands were full manufacturing enough of the larger memory variants. Les Pounder has got his hands on one, of course, and put it through a full testing. Read his Tom's Hardware review: <https://bit.ly/lxf320pi2gb>



All the power, not all the memory.

CREDIT: <https://moonhack.com>

CREDIT: Pineboards

Elegoo Saturn 4 Ultra

It turns out that **Denise Bertacchi** will go crazy for anything that tilts!

SPECS

Build: 218.8 x 122.8 x 220mm

Screen: 10.1-inch monochrome, 12K (11,520x5,120)

Light: COB light source + Fresnel collimating lens

X/Y res: 19 x 24 microns

Z-axis: 0.02mm

Max speed: 150mm/h

Exposure: 2.5 seconds

Control: 4-inch touch panel

Comms: USB, Wi-Fi

Size: 327.4 x 329.2 x 548mm, 14.5kg

It seemed that manufacturers didn't know how to meaningfully improve resin printers without burdening the user with speciality resins and costly consumable parts. Elegoo has realised that it doesn't need to make its printers bigger, faster or higher-res. What we need is a printer that's easier to use.

The Saturn 4 Ultra delivers, with fast, crispy 12K prints using ordinary resin and the same PFA films we've seen on Saturns for years. The improvements are within the machine itself: auto-levelling with built-in sensors, a print monitoring camera that can also shoot time-lapses, and a vat that peels each layer off the film by tilting. Tilting!

The vat isn't the only thing that tilts; so does the hood. This protects uncured resin from ambient UV exposure and is on a hinge that can be tilted back with one hand. No more grabbing the lid with two hands and trying to find a place to set it down. This may seem like a small detail, but it's a big help, especially in limited work space.

The Saturn 4 Ultra needs the same safety precautions as other resin printers. Uncured resin is dangerous, and the solvents used to clean your prints can irritate the skin. Use gloves and goggles when pouring resin and handling uncured prints in a well-ventilated room.

The printer comes fully assembled. You only need to remove the packing materials, and you're good to go. Unlike other resin printers, the Elegoo Saturn 4 Ultra doesn't require manual calibration. It has a mechanical sensor that detects how well the plate and glass fit together and can adjust accordingly.

It's a smart printer with several auto features to assist you. It has sensors to make sure you have enough resin, and it can also tell if bits of a failed model are still stuck to the vat to prevent damage. The AI camera is a mixed bag – it can detect an empty plate, but only once the plate is above the vat, which is around 50mm. It also has warp detection, but it has to be very obvious.

The Saturn 4 Ultra has 12K resolution (11,520x5,120) on a 10.1-inch monochrome LCD screen. This overwhelming number of pixels achieves a 19-micron accuracy on the X axis, which is incredibly crisp, especially when you consider that a human hair averages



The Elegoo Saturn 4 Ultra includes everything you need to get started as soon as you unbox the printer.

70 microns. We've reached a point of resolution where layer lines simply disappear.

The build plate is laser-etched, which holds prints well while still being fairly easy to scrape off. The top of the plate isn't quite sloped enough to get all the resin back into the vat. However, it comes with a drip tray to wrap around the vat to contain resin droplets, and the vat has a no-drip pour spout for getting resin back in the bottle.

The Saturn 4 Ultra ships with a free copy of *Chitubox Basic*, which does everything you need. It handles supports automatically, and has tools for hollowing the model and adding a drainage hole to conserve resin.

We use the same model – Wekster's Rocket – for resin printer tests. This was printed in 8K Standard resin using the default *Chitubox* settings, a 0.05mm layer height, and a 3.5-second layer exposure time. This print took three hours and 13 minutes, which is average. The results are excellent, with clean and clear details around Rocket's teeth and the vines on Baby Groot.

This is an astounding machine with a remarkable level of detail and generous build volume. At 19 microns of resolution, we've hit a wall on the amount of detail we can appreciate with the naked eye. Elegoo finally made a choice to improve the printer itself and not just the light source, giving us a built-in camera, convenient hinged hood and a tilting vat that increases print speed without the need to get extra fine (and expensive) resin. **LXF**

Wekster's Rocket bust – check out the amazing detail on the teeth.



VERDICT

DEVELOPER: Elegoo

WEB: <https://uk.elegoo.com>

PRICE: £444

FEATURES 9/10
PERFORMANCE 9/10

EASE OF USE 9/10
VALUE 9/10

Elegoo's Saturn 4 Ultra is impressive, with never-seen-before features and quality prints all at an affordable price.

» **Rating 9/10**

Pi Pico 2

With £5 burning a hole in his pocket, **Les Pounder** splashes out on the new embedded Pi.

SPECS

SoC: RP2350, dual-core Arm Cortex M33 or dual-core RISC-V Hazard3 up to 150MHz
SRAM: 520KB
SSD: 4MB QSPI
Security: Arm TrustZone, 8KB OTP, Secure Boot
Comms: USB 1.1 device and host
Support: MicroPython, CircuitPython, C, C++
GPIO level: 3.3V
GPIO: 26x digital IO, 4x 12-bit ADC (analogue pins), 2x UART, 2x I2C, 2x SPI, 24x PWM
IO: 12 PIO State Machines
LED: GPIO 25
Power: 1.8-5.5V via micro USB or VSYS
Sleep: <10uA
Size: 21x51mm

It's four and a half years on from the original Pico release back in 2021, and while it may look very much like the original Raspberry Pi Pico, the Pico 2 is powered by a new system on chip (SoC) that sees two CPUs in one package. The RP2350 is a higher-performing microcontroller that features a dual-core Arm Cortex M33 and an open-hardware dual-core RISC-V Hazard3 CPU – more on this later.

Let's address the elephant in the room: the Raspberry Pi Pico 2 does not have any form of Wi-Fi or Bluetooth connectivity. The good news is that we have confirmed with Upton that the Raspberry Pi Pico W 2 is coming later in 2024.

This isn't a full review either – at the time of writing, the software that supports the Pico 2 is not quite ready. MicroPython support for the RISC-V CPU is not yet ready, and our MicroPython firmware is still a preview image. CircuitPython development is ongoing and we have seen the development team working away in the GitHub repositories to make it work.

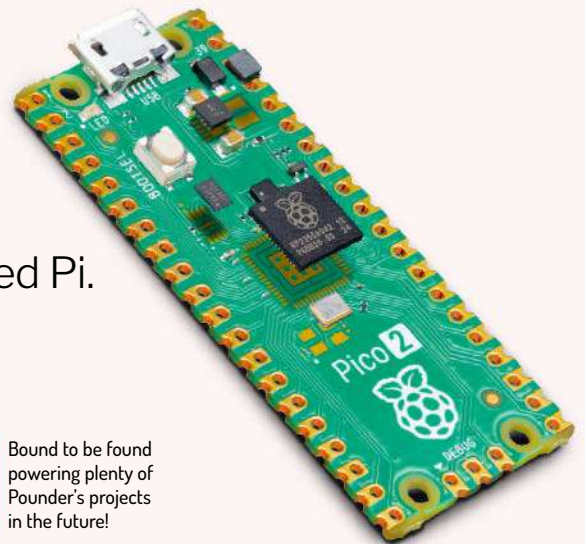
The Pico 2 has roughly double the RAM of its predecessor, and double the flash storage, too. Flash storage is still kept separate from the SoC, but there is 8KB of on-chip storage, reserved for the new Arm TrustZone feature. The stock CPU speed sees a 17MHz boost over the original Pico, but you can overclock the Pico's CPU with just two lines of MicroPython. The real performance boost is in how the Arm Cortex M33 performs versus the older Cortex M0+ and that still remains to be seen.

The Pico 2 uses the same micro USB connector for power and data connectivity. We would have preferred the more robust USB type C connector. Using the same connector and GPIO pinout as the previous Picos makes sense. It means that products designed around the former boards should just work with the Pico 2.

Security blanket

The RP2350 now provides a security architecture that is built using the Arm TrustZone for Cortex-M. There is a signed boot, for instance, 8KB of antifuse OTP that can be used for key storage, SHA-256 acceleration, hardware TRNG (True Random Number Generator) and fast glitch detectors. What does this all mean? It means that if you need additional security features in your project or product, the RP2350 has them baked into the hardware.

The RP2350 has two, dual-core CPU brains. The first is the Arm Cortex-M33, which also utilises the Arm TrustZone framework. The second is the RISC-V Hazard3, the first RISC-V CPU to feature in a Raspberry Pi product. We may not see the open source CPU appear in the main range of Raspberry Pi boards, but it does indicate a possible direction for Raspberry Pi to follow in future Pico boards.



Bound to be found powering plenty of Pounder's projects in the future!

The RISC-V Hazard3 CPU should offer performance somewhere between the original Arm Cortex M0+ of the Pico and the new Arm Cortex M33 of the Pico 2. However, we can't quantify where it falls right now due to a lack of benchmarking software.

According to Raspberry Pi, the RISC-V CPU is available via C, but not via MicroPython. We only had a limited amount of time to write this review, so we have yet to properly test the Pico SDK with C. According to the currently-private GitHub repository, one can pass the 'platform' to CMake in order to build your project UF2 file for the Arm or RISC-V CPUs.

OK, some better news now. Because the Pico and Pico 2 share the same 40-pin GPIO pin layout, we can use add-ons designed for the older boards, with just one catch. Software. If you are wishing to connect I2C, SPI, UART or other types of add-ons, you should be good to go. We connected an OLED display (128x64 pixels) to the I2C interface, installed the SSD1306 module, and were able to send text and images to the screen. That was simple.

If your add-on is a little more exotic – for example, one of Pimoroni's packs – then you need the modules for that particular board. We chose the Pico Display 2.0 and hit an issue: there were no modules in the public GitHub repository. Pimoroni kindly provided a prerelease version of its MicroPython firmware and we can confirm that Pico Display 2.0 works with the Pico 2. Pimoroni will be releasing an updated version of its MicroPython bundle and it should be available very soon, if not already available for download as you read this. **LXF**

VERDICT

DEVELOPER: Raspberry Pi Foundation
WEB: www.raspberrypi.org
PRICE: £4.80

FEATURES	8/10	EASE OF USE	7/10
PERFORMANCE	9/10	VALUE	9/10

Would we buy a Raspberry Pi Pico 2 right now? For £5 sure – the software will catch up and the bugs will be ironed out.

» **Rating TBC**

CREDIT: Raspberry Pi Foundation

PI PICO 2

Hack Wi-Fi on to your new Pi Pico 2

The new Pico 2 doesn't come with Wi-Fi, but when has that ever stopped **Les Pounder**? Never, that's when.



OUR EXPERT

Les Pounder can be found on the Tom's Hardware Pi Cast chatting about Raspberry Pis day in day out.

Just like the original Raspberry Pi Pico, the new Raspberry Pi Pico 2 has no Wi-Fi or Bluetooth connectivity. There will be a Pico 2 W, which has been confirmed by Raspberry Pi co-founder and CEO Eben Upton, but it is still a few months off yet. So, what can we do in the meantime?

Using an Adafruit Airlift FeatherWing, we can hack in wireless internet access. This board is essentially an ESP32, an already formidable Wi-Fi-enabled microcontroller. You may be asking, "Why?" Because we can – and there may be an edge case that requires the power of the new RP2350 via a Wi-Fi connection.

To demonstrate how to use data from the internet, we'll create a simple and fun project that downloads a humorous Chuck Norris 'fact' using an API (application programming interface) that formats the returned data into a format called JSON.

with your Adafruit Airlift FeatherWing. We chose to solder the female header with extra long male header pins to sit in the breadboard.

In the table, pay careful attention to the positions of MISO, MOSI and SCK. The pinout for the FeatherWing sees more pins on one side than the other. Check and double-check before powering on.

Pico 2	FeatherWing
VSYS	USB
GND	Any GND pin
GPIO 10	SCK
GPIO 11	MOSI
GPIO 12	MISO
GPIO 13	ESPCS
GPIO 14	ESPBUSY
GPIO 15	ESPRST

YOU NEED

- > Pico 2 or Pico 1
- > Adafruit Airlift FeatherWing co-processor
- > 8x M2M jumper wires
- > Breadboard

Featherweight

Adafruit's Feather range of boards use a different pinout to the Raspberry Pi Pico 2, so we'll need header pins soldered to the Raspberry Pi Pico 2 and to the FeatherWing Airlift (www.adafruit.com/product/4264). As we are using both boards in a breadboard, the included male headers will suffice.

If you intend to use the FeatherWing with an Adafruit Feather board such as the Feather RP2040, ensure that you need to have female headers soldered to the top of the microcontroller. These should come

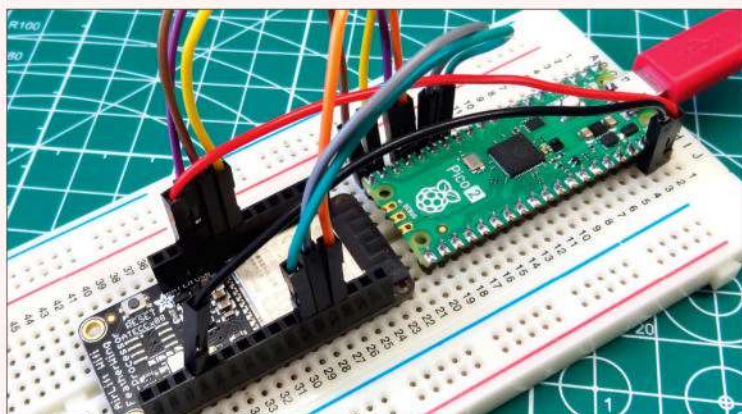
CircuitPython in a flash!

Flashing CircuitPython to a microcontroller is easy – it was designed to be. We're going to be using an alpha release of CircuitPython for the Raspberry Pi Pico 2; there may be issues and crashes, so we wouldn't deploy this into a mission critical project. Wait for a full release if that is the case.

Download the latest version of CircuitPython (<https://bit.ly/lxf320cp>) for the Pico 2. At the time of writing, this was an alpha release, so beware the occasional crash. Press and hold the BOOTSEL button on the Pico and insert a USB cable into the Pico and your computer.

Copy the CircuitPython UF2 file to the **RP23502** drive. The drive will unmount and disappear while flashing the code. A new drive **CIRCUITPY** will appear, confirming that the flash was a success.

Download the libraries archive (<https://circuitpython.org/libraries>) for the version of CircuitPython that you have downloaded. Extract the contents of the download to a folder and copy the following files/folders to the **lib** folder on the **CIRCUITPY** drive: `/adafruit_bus_device`



Not quite as neat as having built in Wi-Fi, that's for sure!

```
/adafruit_esp32spi
Adafruit_requests.mpy
```

Writing the project code

Our project is written in CircuitPython, a version of MicroPython from Adafruit. CircuitPython differs by offering an easy way to install modules of code for use with Adafruit's range of products. We're going to use *Thonny* to work with CircuitPython code that will communicate with the Adafruit FeatherWing to pull data from the web.

Download and install *Thonny*, then go to Tools > Options and select the Interpreter tab. Set the Interpreter to CircuitPython (Generic) and the Port to match the USB serial device (the Pico 2). Click OK. You can also try the auto port option to force *Thonny* to search for the port.

Click on STOP to force the Interpreter to connect and run the Python shell on the Raspberry Pi Pico 2. Our first block of code imports a series of CircuitPython modules in order to create the project:

```
import board
import busio
from digitalio import DigitalInOut
import adafruit_connection_manager
import adafruit_requests
from adafruit_esp32spi import adafruit_esp32spi
from secrets import secrets
```

Print a message to the Python shell, telling the user what the project is about:

```
print("Raspberry Pi Pico 2: Chuck Norris Joke-
Randomizer")
```

Create an object to store the URL of the Chuck Norris jokes. This can be the URL of any API or JSON source that you wish to use:

```
CHUCK_NORRIS_URL = "https://api.chucknorris.io/
jokes/random"
```

Now set up the GPIO pins used for making the SPI connection between the Raspberry Pi Pico 2 and the Adafruit FeatherWing:

```
esp32_cs = DigitalInOut(board.GP13)
esp32_ready = DigitalInOut(board.GP14)
esp32_reset = DigitalInOut(board.GP15)
spi = busio.SPI(board.GP10, board.GP11, board.GP12)
esp = adafruit_esp32spi.ESP_SPIcontrol(spi, esp32_cs,
esp32_ready, esp32_reset)
```

Set up the networking infrastructure aspect of the code – this is all the behind-the-scenes tech that makes the connection possible:

```
pool = adafruit_connection_manager.get_radio_
socketpool(esp)
ssl_context = adafruit_connection_manager.get_radio_
ssl_context(esp)
requests = adafruit_requests.Session(pool, ssl_context)
```

Print a message to say that the project will attempt to connect to your Wi-Fi:

```
print("Connecting to AP...")
```

While the ESP32 in the Adafruit FeatherWing is not connected, the code tries to make a connection using the Wi-Fi AP and password stored in the **secrets.py** file. We will make this file later.

```
while not esp.is_connected:
```

```
try:
```

```
    esp.connect_AP(secrets["ssid"],
secrets["password"])
```

Using these lines, handle any connection errors. If there is an error, the code keeps trying to connect.

```
except OSError as e:
```

```
    print("could not connect to AP, retrying: ", e)
    continue
```

Print the connection details for a successful connection. These lines print the access point name, the internal IP address of the project, and an initial message to announce the random Chuck Norris joke:

```
print("Connected to", esp.ap_info.ssid, "\tRSSI:", esp.
ap_info.rssi)
print("My IP address is", esp.ipv4_address, "\n")
print("Your Random Chuck Norris joke is... ")
```


Using Adafruit's requests, get the latest joke in a JSON format and store it in an object called **r**. Extract the joke from the JSON object. JSON is similar to Python's dictionary data structure, which uses keys to extract values from the object. Here the key is called **value**, which may be confusing but we can't control it.

```
r = requests.get(CHUCK_NORRIS_URL).json()
joke = r['value']
```

Print the joke, encapsulated inside of a frame made from **"-"**. Using **len()**, we can get the length (number of characters in the joke string) and use it to create the frame with the correct number of characters.

```
print("-" * len(joke))
print(joke)
print("-" * len(joke))
```

Save the code to the Pico 2 as **code.py**. This will overwrite the existing **code.py** file and each time the Pico 2 powers up, the code autostarts.

We are now ready to learn some pearls of wisdom regarding Chuck Norris. Go back to the **code.py** tab. Click on Run > Run Current Script (or press F5, or the green play button) to start the code. The code runs in the Python Shell, and after a few moments it should connect to your Wi-Fi and download the latest Chuck Norris joke using the JSON API. 

» CREATING A SECRETS FILE

The **secrets.py** file is a useful tool that stores our Wi-Fi details, along with country-specific details that can be used for Wi-Fi connections. By keeping our secrets separate from the main code, we can share the **code.py** file without fear of sharing our secrets.

1. Create a new file on the Raspberry Pi Pico 2.

2. Create an object called **secrets** and store your Wi-Fi SSID, password and time zone inside the object. This object is a Python dictionary. The keys identify what the data stored in the values represents.

```
secrets = {
    'ssid' : 'YOUR SSID',
    'password' : 'YOUR PASSWORD',
    'timezone' : 'YOUR TIMEZONE SEE THE URL >> ', # http://
worldtimeapi.org/timezones
}
```

3. Save the file to the Raspberry Pi Pico 2 as **secrets.py**.

» GET YOUR Pi FILLING HERE Subscribe now at <http://bit.ly/LinuxFormat>

BREADBOARDOS

Credit: <https://github.com/mcknly/breadboard-os>

Cut a slice of Pi Pico on BreadboardOS

Filling up on tasty carbs is one way **Tam Hanna** spends his weekends, when not making questionable devices in his underground lab.



**OUR
EXPERT**

Tam Hanna has seen a wide array of real-time operating systems. This gives him the necessary experience to analyse this new market entrant.

As the demands placed on embedded systems have become ever more complex, real-time operating systems have evolved into integral parts of embedded system design. Amazon's decision to purchase Real Time Engineers (the company behind FreeRTOS) injected venture capital into what was previously a somewhat tranquil market dominated by professional nerds.

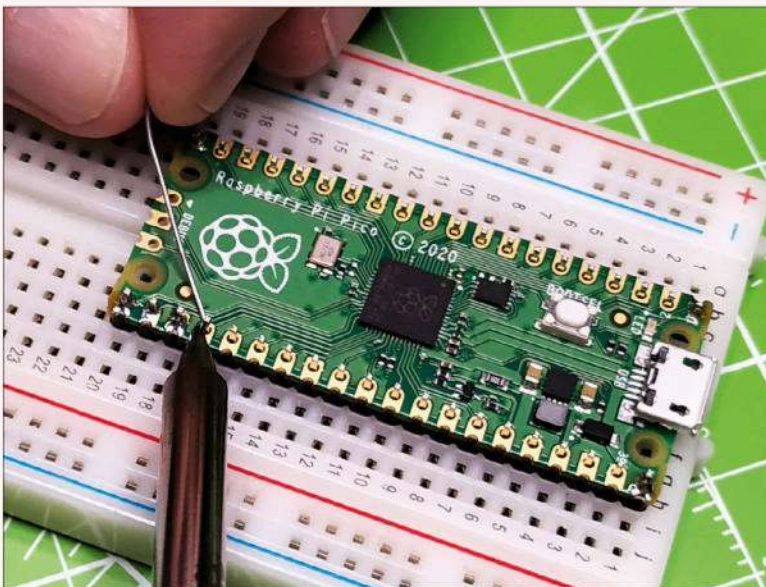
Real-time operating systems, however, share the name with the packages of software commonly considered an operating system. While Windows, Linux, and Mac OS are focused on providing a wide field of user-facing parts, a real-time operating system's *raison d'être* is shuffling around computer resources and tasks, ideally by providing a guaranteed response latency to interrupts and other events. User-facing components are not a central part of the equation, with command-line interfaces and similar comfort features being considered a complete waste – a real-time operating-system-based application is a monolith, where the operating system and application code are compiled into one ROM image.

BreadboardOS aims to shake this up. While it is based on FreeRTOS, the development team has added command-line utilities intended to make developers' lives easier. Given that the operating system works well on the Raspberry Pi Pico, let's look at what the latest contender in the real-time operating system market has to offer.

Start wrapping

Given that BreadboardOS is a wrapper around FreeRTOS, parts of the installation and deployment process should not be completely new to developers experienced with the system.

The following steps occur on an Ubuntu 20.04 LTS-based workstation – compiling the operating system



Real-time OSes are designed to run on embedded devices, so break out your spare Pi Picos!

using WSL is also possible but not recommended due to problems when interacting with serial hardware. Be that as it may, you should first check for the presence of relevant packages:

```
$ sudo apt-get install build-essential gcc-arm-none-eabi newlib-arm-none-eabi cmake
$ sudo apt install cmake gcc-arm-none-eabi libnewlib-arm-none-eabi libstdc++-arm-none-eabi-newlib
```

Installing the library `newlib-arm-none-eabi` sometimes fails. If `apt-get` complains about errors related to this library, it can be ignored without peril.

Base filesystem

Real-time operating-system-based applications are combined into a package with application source code. Starting the compile process commences by erecting the file structure and pulling the relevant components from their respective GitHub repositories:

```
$ mkdir bbosspace
$ cd bbosspace/
$ git clone https://github.com/FreeRTOS/FreeRTOS-Kernel.git
$ git clone https://github.com/raspberrypi/pico-sdk.git
```

```
[ 5950.638031] usb 1-1.4: new full-speed USB device number 7 using ehci-pci
[ 5950.858742] usb 1-1.4: New USB device found, idVendor=2e8a, idProduct=000a, bcdDevice= 1.00
[ 5950.858748] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 5950.858752] usb 1-1.4: Product: Pico
[ 5950.858755] usb 1-1.4: Manufacturer: Raspberry Pi
[ 5950.858757] usb 1-1.4: SerialNumber: E6614C311B3B5E36
[ 5950.913449] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
[ 5950.918999] usbcore: registered new interface driver cdc_acm
[ 5950.919000] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
tamhan@TAMHAN18:~/bbospace$
```

The Raspberry Pi Foundation's Pico SDK consists of a set of network repositories. Due to that, it is mandatory to change into its working directory and execute a submodule update run. Failing to do so will lead to non-working project files and must be avoided at all costs:

```
$ cd pico-sdk/
$ git submodule update --init
```

The BreadboardOS build environment uses two local variables to keep track of the paths to the FreeRTOS kernel and the Raspberry Pi Pico SDK. If you do not want to modify the configuration of your workstation, enter the following two commands into a terminal window to prime it for compilation. Be careful, however, not to forget the environment variables, otherwise the tools will fail:

```
$ export FREERTOS_KERNEL_PATH=/home/<your username>/bbospace/FreeRTOS-Kernel
$ export PICO_SDK_PATH=/home/<your username>/bbospace/pico-sdk
```

The final act of the initialisation involves downloading the Breadboard OS wrapper files. It is also a modular project, which is why an invocation of *submodule update* is needed after the main GitHub clone process runs through:

```
$ cd ~/bbospace
$ git clone https://github.com/mcknly/breadboard-os.git
$ cd breadboard-os/
$ git submodule update --init
```

Deploy an image

CMake's build system achieved a tremendous amount of mind share in the embedded space during the last few years. As in the case of most CMake projects, a file called **project.cmake** controls the compilation process. Fortunately, the BreadboardOS standard distribution provides one of these, which can be opened by entering **gedit project.cmake** in the folder **~/bbospace/breadboard-os**.

Given that the following steps will be done on the Raspberry Pi Pico W, the following two changes are first required:

```
# CLI INTERFACE - 0: use UART for CLI (default), 1: use
USB for CLI
set(CLI_IFACE 1)
...
# BOARD - set the board being used (i.e. 'pico', 'pico_w',
etc)
set(BOARD pico_w)
```

Firstly, it's crucial to set the **CLI INTERFACE** to one. This configuration enables the virtual USB interface to expose the BreadboardOS command-line interface. Secondly, the **BOARD** attribute needs to be set up to ensure the correct behaviour of the HAL.

After that, commands need to be entered in order to start the compile process. The *mkdir* invocation is required only for the first time – it primes the filesystem and creates space for the relevant build files:

```
$ cd ~/bbospace/breadboard-os
$ mkdir build && cd build
$ cmake ..
$ make -j8
...
[100%] Built target my-bbos-proj
```

When done, a UF2 file is created. If you did not modify the project attributes in the build file, the file name will be **my-bbos-proj.uf2**. Be that as it may, the next step involves holding the BOOTSEL button while plugging the board into USB in order to enable the Raspberry Pi Foundation bootloader. After that, the UF2 file containing the application needs to be deployed just as though it were a standard Raspberry Pi Pico application.

After the obligatory restart, the Raspberry Pi Pico shows up in *dmesg* similar to the way shown in the screenshot (above).

In the next step, a TTY connection is required for experimentation with the CLI. Given that this tutorial is not about terminal emulators, use the screen to get access to an interface similar to the one shown in the second screenshot (over the page):

```
~/bbospace$ screen /dev/ttyACM0
```

As the screen takes complete control of the keyboard, exiting is a little difficult. First hit Ctrl+a, and then press the keys **k** and **y** to retake control of the terminal window.

This Raspberry Pi Pico is ready to receive commands.

QUICK TIP

In code sections, we use the **\$** to denote terminal commands you need to enter. Text before the **\$** shows you what folder you're expected to be in, if we haven't explicitly coded into one.

» WHAT IS BREADBOARDOS?

Also known as BBOS, BreadboardOS provides tools that make developing projects much easier. Its command-line interface (or CLI) lets you interface with the RP2040 board directly. This makes it possible to not only interact with the Pi but also handle practical functions like real-time debugging. This can be a time-saver when troubleshooting, getting you closer to your end project faster.

Developer Cavin McKinley explains more about the development of BBOS on the official project page. BBOS was developed from scratch by McKinley to help not only with his projects but also those of others as an open source platform. It was created on top of FreeRTOS but he says that experience with FreeRTOS isn't required.

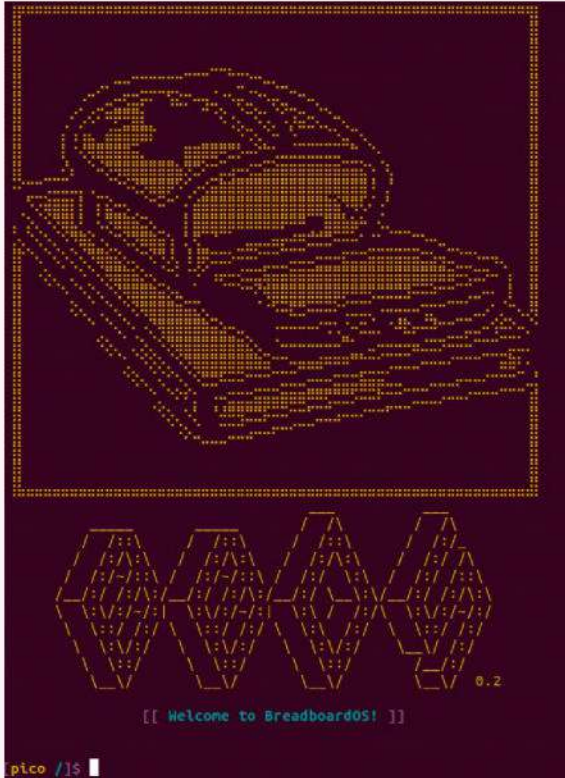
BreadboardOS is described by McKinley as a microshell platform. The CLI lets you interact with the hardware but also sorts data into a hierarchy of folders that, in theory, should be easy to pick up on and use. If you're looking for a new way to develop RP2040 projects, there are detailed instructions provided over at GitHub that explain how to download and install BBOS.



The BreadboardOS command line is ready to receive commands.

QUICK TIP

Because BreadboardOS is an abstraction layer on top of FreeRTOS, we recommend Warren Gay's *FreeRTOS for ESP32-Arduino*, available from Elektor. It's a comfortable-to-use and easy-to-read introduction.



Finally, the CLI sometimes needs a special invitation. If you establish a new connection to the Raspberry Pi Pico and do not see the prompt `[pico /]$`, press Return once or twice to startle the system into operation.

Pi on the Breadboard

Once the screen terminal emulator connects to the Pi Pico, it is time to experiment with the features of the command line. Firstly, enter `help` to show a list of commands (see screenshot below). Invoking `help` inside another folder, such as `bin`, changes the display.

Most interestingly, the Raspberry Pi Pico's remanent storage is populated with directories. This can best be proven by entering the `ls` and `cat` commands as per the lines below:

```
[pico /]$ ls
d---- .
d---- mnt/
d---- etc/
d---- bin/
d---- proc/
d---- dev/
```

The BreadboardOS CLI comes with several useful commands.

```
[pico /]$
[pico /]$ help
clear          - clear screen
help          - list available commands
ls            - list directory content
cd            - change current directory
pwd           - print current directory
cat           - print files content
xxd           - dump file hex content
echo          - print string to file
```

```
d---- lib/
-r--- info.txt
[pico /]$ cat info.txt
my-bbos-proj, compiled with love
```

One very interesting aspect involves the presence of the `dev` folder. Invoking the `ls` command displays the results shown (see screenshot opposite). Most of the virtual devices can be interacted with in order to enable direct interaction with the underlying hardware.

Interestingly, the current active folder is shown in a fashion similar to `[pico dev]` – the path structure of BreadboardOS differs from Linux or Windows.

Start slicing

Now it's time to proceed to the creation of the first BreadboardOS task. Given that our platform is based on FreeRTOS, the individual applications hosted by the operating system are, of course, based on the design paradigms known from the underlying platform.

The tasks living inside a BreadboardOS installation are called services. In a freshly downloaded version of the OS, the `services` folder comes with a small template, which we will expand on to find out more.

The CLI mentioned above provides support for exploration. When changing the current directory to `bin`, the following utilities are made available:

```
[pico /]$ cd bin
[pico bin]$ ls
d---- .
d---- ..
--x- ps      - print running service info
--x- top     - print runtime stats for services
--x- free    - print heap memory (RAM) usage stats
--x- df      - print flash memory usage stats
--xh kill    - kill the service name given by 'bin/ps'
--xh service - interact with available services
--xh reboot  - reboot device
```

In particular, the `service` command permits the starting and stopping of various payloads – entering the command `services list` provides an overview of all services found in the current BreadboardOS instance. The following steps look at the file `breadboard-os/services/heartbeat_service.c` in some detail – it serves as a base for our custom service.

In addition to the code, the file `bbospace/breadboard-os/services/services.h` also is relevant – it acts as a hub for all the services contained in the app.

Given that our new service is based on its heartbeat colleague, the following constants are essential:

```
#define SERVICE_NAME_HEARTBEAT heartbeat
#define PRIORITY_HEARTBEAT 1
#define REPEAT_HEARTBEAT 1
#define DELAY_HEARTBEAT 5000
// Example heartbeat service "beats"
// every 5 seconds when started
#define STACK_HEARTBEAT configMINIMAL_STACK_SIZE
```

In addition to values determining the priority the real-time operating system assigns to the task, we also find the attribute `DELAY_HEARTBEAT`. It determines the tick rate at which the real-time operating system should re-activate the job.

One further topic involves memory allocation: in systems with high security demands, dynamic memory is frowned upon as it has the propensity to fail. Due to that, each thread's stack must be pre-allocated – passing in the constant `configMINIMAL_STACK_SIZE` tells FreeRTOS that this thread can run with minimal amounts of memory.

With this out of the way, we can return to the code file. It contains the invocation method `heartbeat_service(void)`, which spawns the real-time operating system task according to the following lines:

```
BaseType_t heartbeat_service(void)
{
    BaseType_t xReturn;

    xReturn = xTaskCreate(
        prvHeartbeatTask,    // main function of this
                             // service, defined below
        xstr(SERVICE_NAME_HEARTBEAT), // name
                                     // defined in services.h
        STACK_HEARTBEAT,    // stack size defined in
                             // services.h
        NULL,                // parameters passed to created
                             // task (not used)
        PRIORITY_HEARTBEAT, // priority of this
                             // service, defined in services.h
        &xHeartbeatTask      // FreeRTOS task handle
    );
```

Developers experienced with FreeRTOS will immediately recognise the method `xTaskCreate` – it also spawns non-BreadboardOS tasks. The actual intelligence comes in an endless loop similar to this:

```
static void prvHeartbeatTask(void *pvParameters)
{
    static char *heartbeat_string = "ba-bump";
    while(true) {
        cli_print_timestamped(heartbeat_string);

        task_sched_update(REPEAT_HEARTBEAT, DELAY_HEARTBEAT);
    }
}
```

`Task_sched_update` informs the kernel that the task has finished its current job and wants to return control to the operating system scheduler.

This is important, because badly behaving tasks can lead to starvation of the idle thread – in this case, FreeRTOS's internal state will eventually become undefined because there are no compute resources available for housekeeping tasks.

Blinking LEDs

Our decision to base this tutorial on the Raspberry Pi Pico W actually becomes problematic here, because the wireless module partially handles this board's GPIO complement.

One particular pain point concerns the LED. Due to this, an external multimeter or a light-emitting diode and a resistor connected to GPIO pin 16 will be used as the output device.

Next, return to the file acting as the host for the main service. Change the worker method as following:

```
[pico /]$ cd dev
[pico dev]$ ls
d---- .
d---- ..
-r-- time           - system timer
-r-xh gpio          - GPIO pins
-r-xh i2c0          - I2C bus 0
-r-xh spi0          - SPI bus 0
-r-- adc0           - Analog-to-Digital Converter
-rw- uart1          - auxilliary UART
```

```
#include "pico/stdlib.h"

static void prvHeartbeatTask(void *pvParameters)
{
    static char *heartbeat_string = "ba-bump";
    gpio_init(16);
    gpio_set_dir(16, GPIO_OUT);
    char state = 0;

    while(true) {
        cli_print_timestamped(heartbeat_string);
        gpio_put(16, state);
        if(state==0)state=1;
        else state=0;
    }
}
```

The code here is simply a reimplementaion of the well-known *blink* program. After configuring the various GPIO elements, we change into a loop in which we permanently toggle the state of the pin.

After that, trigger a CMake rebuild using the commands mentioned above. When done, redeploy the UF2 file to the process computer and return to screen to start the program as per the following:

```
[pico /]$ cd bin
[pico bin]$ service start heartbeat
[pico bin]$ heartbeat service started
[pico bin]$ [26716387 ] ba-bump
[pico bin]$ [31715705 ] ba-bump
```

Even though BreadboardOS is a wrapper around FreeRTOS, it is an exciting experimentation environment ideal for people wanting to learn more about real-time operating systems. **LXF**

BreadboardOS provides access to various hardware features.

QUICK TIP

In our tests, a Windows workstation was unable to open its Start menu when a Raspberry Pi Pico running BreadboardOS was connected. Remove the BreadboardOS-based Raspberry Pi from the workstation before booting Windows!

» REAL-TIME OPERATING SYSTEMS

Wikipedia defines a real-time OS (RTOS) as one "that processes data and events that have critically defined time constraints. An RTOS is distinct from a time-sharing operating system, such as Unix, which manages the sharing of system resources with a scheduler..." You can consider them as event-driven and are expected to respond within a specific amount of time – typically a short, predictable one. They're usually deployed in embedded systems with a focused job in mind, where general processors aren't required.

Just as with desktop and server operating systems, there's a range of RTOSes available with differing levels of expectations. BreadboardOS is based on FreeRTOS (www.freertos.org), which has been around for 15 years and is available for over 40 embedded devices. Amazon took stewardship of FreeRTOS back in 2017 and added libraries to it for easy access to Amazon Web Services for internet of things devices.

» GET MORE REAL-TIME UPDATES... Subscribe now at <http://bit.ly/LinuxFormat>

Speed up downloads

Optimum use of internet bandwidth is **Shashank Sharma's** life's purpose. Thankfully, there are tools designed to help do just that.



**OUR
EXPERT**

Shashank Sharma is a trial lawyer in New Delhi and an avid Arch user. He's been writing about open source software for 20 years and lawyering for 10.

When you've got a tool to grab torrents and your web browser has a download manager for most other files, do you still need a dedicated download manager? Short answer, yes – use *Aria2*.

Aria2 is a lightweight, multi-protocol, multi-server and multi-source command-line download utility that can verify checksums and authentication certificates on the fly. Furthermore, it includes a fully featured BitTorrent client with support for DHT, PEX, encryption, magnet URIs, web seeding, selective downloads, local peer discovery and a UDP tracker. You can also throttle the bandwidth, so you can watch YouTube while downloading the latest release of your favourite distro.

While this might not be a deciding factor for modern multi-core multi-processor computing devices, CLI utilities like *Aria2* are incredibly lightweight. And no matter how many GHz and cores your computer has, people are always put off by bloatware. And we've saved some of its more impressive features for last.

For one, *Aria2* supports opening multiple connections to download the file even faster. But perhaps its niftiest feature is that it can download the same file at the same time using different protocols.

Released under the GPLv2 licence, *Aria2* is available in the software repositories of most popular desktop distros. If you're running Ubuntu, Debian or a derivative, you can install it with the `sudo apt install aria2` command. Run the `sudo dnf install aria2` command instead if you're on a RPM-based distro.

Speedy downloads

As *Aria2* supports HTTP, HTTPS and even FTP, in addition to magnet and torrent links, the simplest

command to begin downloading a file is `aria2c <url>`. The tool also supports auto-resume if the connection is terminated by the user or for any other reason. For instance, you can stop the download by pressing Ctrl+c, and then resume again with the original download command.

Before beginning the download, *Aria2* allocates disk space for the file to be downloaded. There are various methods, or you can forego the process altogether. Depending on the size of the file you wish to download and the method of disk allocation, this process might take some time. For a ~2GB file, the process took a little under 10 seconds using the default disk allocation method on our test machine. The default method is `prealloc`, which is time-consuming, so you can instead opt for `falloc`, which is much faster and more suitable for modern filesystems such as ext4 and Btrfs. To set the disk allocation method, you must use the `--file-allocation=<METHOD>` command option.

The `aria2c --file-allocation=falloc https://download.fedoraproject.org/pub/fedora/linux/releases/40/Workstation/x86_64/iso/Fedora-Workstation-Live-x86_64-40-1.14.iso` command uses the `falloc` method to assign space before starting the download. You can similarly use the `--file-allocation=none` command option if you don't wish to allocate disk space before starting the download.

Aria2 generates a lot of output, but it pertinently shows the size of the target file, the data downloaded, the speed of the download and the time remaining under the Download Progress Summary heading:

```
*** Download Progress Summary as of Tue Aug 27
15:15:00 2024 ***
```

```
=====
[#5ccdf9 186MiB/2.1GiB(8%) CN:1 DL:4.3MiB
ETA:7m36s]
FILE: /media/linuxlala/Stuffsies/distros/Fedora-
Workstation-Live-x86_64-40-1.14.iso
=====
```

This shows that 186MB has been downloaded from a total of 2.1GB, which is about 8%, and the download speed is 4.3MB and ETA is around 7.5 minutes.

By default, *Aria2* sucks up all available bandwidth to finish the downloads, so you might find the `--max-download-limit=<speed>` command option helpful. The default option is zero, which means no limit, but you

Although the project is named *Aria2*, you launch the CLI utility with the `aria2c` command.

can define the limit in bytes/second. If you have sufficient bandwidth, you can also download multiple files concurrently with `-j<value>`. For instance, to simultaneously download two files, use the `aria2c -j2 <file1 url> <file2 url>` command.

Aria2 also lets you specify multiple sources for the file to be downloaded. For instance, when downloading the latest release of EndeavourOS, we specified both the India mirrors and the Singapore mirror. The output generated by Aria2 showed that it made three connections. But even when downloading from a single source, you can make multiple connections to speed up downloads, using the `-x<num>` command option.

```
$ aria2c -x2 https://mirror.nag.albony.in/endeavouros/iso/EndeavourOS_Endavour-2024.06.25.iso http://mirrors.nxtgen.com/endeavouros-mirror/iso/EndeavourOS_Endavour-2024.06.25.iso https://mirror.freedif.org/EndeavourOS/iso/EndeavourOS_Endavour-2024.06.25.iso
```

```
08/27 16:28:47 [NOTICE] Downloading 1 item(s)
[#5a6a82 737MiB/2.7GiB(26%) CN:5 DL:7.6MiB ETA:4m26s]
```

In this example, even though there are three sources, Aria2 is telling us that it has made five connections (CN:5).

More download options

To download a torrent file, you can either point Aria2 to the downloaded .torrent file or to the URL. In the latter case, Aria2 firsts download the .torrent file and then starts the download. You can similarly point Aria2 to the magnet link to begin the download. When you run the `aria2c --follow-torrent=false "URL.torrent"` command, Aria2 merely downloads the .torrent file, but doesn't start the torrent.

There are still more options to configure the torrents, such as defining the destination directory, specifying a different name for the downloaded files, setting download speeds, throttling upload speeds, defining a seed ratio and more. As is often the case with featureful CLI utilities, the man page is the best source of information for all configurable parameters and available command options.

```
1: linuxlala@playground: /media/linuxlala/Stuffsies/distros
linuxlala@playground: /media/linuxlala/Stuffsies/distros$ aria2c -j2 -i aria-down
load-list.txt
08/27 18:07:47 [NOTICE] Downloading 2 item(s)
08/27 18:07:48 [NOTICE] File already exists. Renamed to /media/linuxlala/Stuffsies/distros/Fedora-Astronomy KDE-Live-x86_64-38.1.torrent.
DL:167KiB[#57ff90 16KiB/378KiB(4%)]#487ab4 16KiB/172KiB(9%)
08/27 18:07:49 [NOTICE] Download complete: /media/linuxlala/Stuffsies/distros/Fedora-Cinnamon-Live-x86_64-38.torrent
08/27 18:07:49 [NOTICE] IPv4 DHT: listening on UDP port 6914
08/27 18:07:49 [NOTICE] Allocating disk space. Use --file-allocation=none to disable it. See --file-allocation option in man page for more details.
08/27 18:07:49 [NOTICE] Download complete: /media/linuxlala/Stuffsies/distros/Fedora-Astronomy KDE-Live-x86_64-38.1.torrent
DL:0B[#aald36 0B/2.1GiB(0%)]#0a874a 0B/4.7GiB(0%)[FileAlloc:#aald36 2.0GiB]
08/27 18:08:13 [NOTICE] IPv4 BitTorrent: listening on TCP port 6940
```

When you have a large number of files to download, instead of adding them all to a single command, you can instead create a text file, with each file to be downloaded relegated to its own line. On each line, you can provide a tab-separated list of different sources for the same file. You can also define various parameters for each download in a separate line:

```
https://file1      https://mirror/file1
dir=/path/to/destination/directory/
out=output-filename
https://some-URL.iso
https://torrent-link
--max-upload-limit=1M
```

Save this to a file, and then point Aria2 to it with the `aria2c -i filename.txt` command. In our example above, you can see we've added three different files to download. For the first, we've provided two sources, as well as the destination directory and an output filename. For the second file to be downloaded, we haven't provided any options, and for the third, which is a torrent, we've specified the maximum upload limit. The options must be placed below the URLs in a new line that begins with a single white space.

Aria2 is one of the most featureful and well-documented CLI utilities we've worked with in some time. You'll quickly realise how much you needed it once you try it for yourself. **LXF**

If you don't use the `-j` command option, Aria2 will wait until the first file is downloaded before downloading the next.

QUICK TIP

The advantage of using multiple sources, when available, is that if one fails for any reason, Aria2 continues the download from the second without requiring user intervention.

» GRAPHICAL FRONT-ENDS

As useful as CLI utilities are, some users are more comfortable with graphical apps. If you like the feature set of Aria2 but not the notion of working with the CLI to manage downloads, you can try you either *Varia* or *AriaNg*.

Varia offers a Flatpak file for quick installation, but if you don't want a desktop app, you can use the *Firefox* or *Chrome* extension. The

project is written in Python and has only a handful of dependencies, all of which can be satisfied from your distro's software repositories. You'll find installation instructions on the project's GitHub page (<https://github.com/giantpinkrobots/varia>). Although not as featureful as the parent CLI utility, you can still resume paused downloads, define download/upload speeds, authenticate

with username/password, launch simultaneous downloads, and more. While *Aria2* is fully cross-platform, *Varia* only offers experimental support for Windows.

Another option is the *AriaNg* project, which is built using HTML and JavaScript, and you only need a browser to make use of all the features of *Aria2*. More importantly, the project

boasts a responsive design, and features a dark theme. Head over to the project's GitHub page (<https://github.com/mayswind/AriaNg>) for installation instructions. The project also offers a variant, *AriaNg-Native* (<https://github.com/mayswind/AriaNg-Native>), which provides a desktop utility, but one that supports almost all the features of *Aria2*, and certainly more than *Varia*.

» **ENHANCE YOUR TERMINAL-FU** Subscribe now at <http://bit.ly/LinuxFormat>

LINUX BASICS

Part Six!
Don't miss
next issue,
subscribe on
page 16!

Fix and tweak your network settings

Nick Peers uncovers the mysteries of how to set up, troubleshoot and improve network performance in Ubuntu and other distros.



**OUR
EXPERT**

Nick Peers can't believe how far networking has come in Linux in his 20-odd years using Ubuntu. You youngsters don't know how lucky you are...

These days, connecting your PC to the network is usually a simple plug and play affair – especially when connecting via Ethernet cable. Each iteration of Ubuntu – particularly when you jump from one LTS version to the next – refines things that little bit more, and it's all too easy just to jump straight back into computing and forget about the technology that enables you to surf the internet and add new software with just a few clicks. But what happens if something goes wrong or you need to make changes to your network setup? Read on to discover how to access Ubuntu's networking tools and troubleshoot problems with your Wi-Fi connection – including poor performance due to network congestion from your neighbours.

Network settings

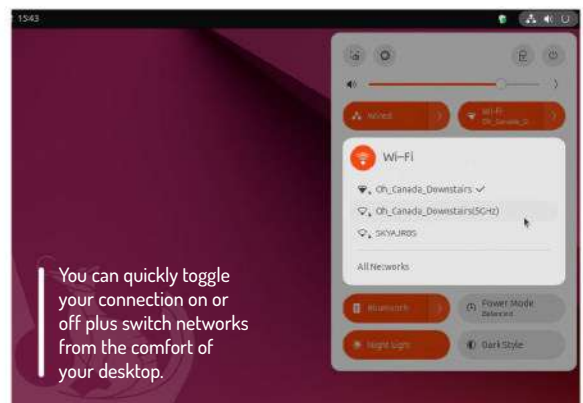
Your network is front and centre on the Ubuntu desktop – it's one of the three system icons that appears in the top-right corner of the Gnome desktop. Click this to bring up the system flyout and you'll see Wired or Wireless buttons (or both) appear. If the connection is enabled and working, it appears orange; otherwise it's greyed out.

Click the button to quickly enable or disable the connection, or click the > button to view all available connections – in the case of Wired, your Ethernet connection will be shown along with a Wired Settings shortcut that takes you to the Network section of Settings. Expanding Wireless reveals a list of all Wi-Fi networks within range, in addition to the one you're currently connected to, enabling you to easily switch. You'll also see an option to access All Networks, which takes you to a separate Wi-Fi section in Settings.

Wired connection

If you click Wired Settings, you're shown a list of connections: Ethernet, VPN and Proxy. Again there are quick-fire on/off toggles for existing connections, but you'll also see a + button next to each for setting up new connections from scratch, plus a settings button to view and tweak aspects of any existing connections.

Let's start with your Wired connection: click the settings button to reveal a multi-tab window opened to the Details tab, revealing key information about your



connection. The link speed refers to the current speed of your connection – if you're connected via a standard 1Gb/s Ethernet cable, it'll be 1,000Mb/s, for example.

Below this are IPv4 and IPv6 addresses – the key one to know is your PC's IPv4 address, which identifies it on your local network. This is a four-digit address, with each digit a number between 0 and 255. In most cases, the first two digits are usually 192.168, while the third digit (often 0 or 1) is the same for all devices on your network. It's the final digit that's unique to each device on your network. This means you can have up to 256 devices (including your router) connected to your local network.

Below the IP addresses is your PC's Hardware Address, also known as the MAC address. This is a six-digit hexadecimal address unique to your device, which provides another means of identifying it elsewhere on your network – through your router's configuration utility, for instance.

Beneath this you'll see a Default Route address, similar to your IPv4 address with one key difference (the final digit). This refers to your router or router/modem's IP address, and the final entry – DNS – may well also be set to your router. The DNS server is used to help navigate your way around the web by translating a web address such as **www.ubuntu.com** or **www.linuxformat.com**, for example, to its four-digit public IPv4 address.

The Details tab also contains three checkboxes, two of which are ticked by default. The Connect

QUICK TIP

When setting a static IP address for your PC, it's important not to choose one already in use by another device. Use a scanning tool such as **Angry IP Scanner** (<https://angryip.org>) to see what IP addresses have been allocated.

Automatically option ensures your PC automatically connects to the network when you boot it up, while the Make Available To Other Users option allows your network connection to be used by other user accounts on your PC. You should disable this option if you want to block access to the network and internet on that network interface.

The final option – Metered Connection: Has Data Limits Or Can Incur Charges – prevents large files such as software updates from being automatically downloaded, regardless of whether or not your connection is actually metered.

Make changes

The other tabs here enable you to tweak certain aspects of your internet connection. Start with the Identity tab, where you can rename the connection to something more memorable. Should you wish to clone or spoof your MAC address for any reason, use the Cloned Address drop-down menu to do so. There are four options here: Preserve keeps the MAC address as it was when previously booted. You should choose Permanent to use your hardware's genuine MAC address, or Random to generate one completely at random. The final option is Stable, which generates a fake MAC address that remains the same whenever the connection is enabled, ensuring your PC can hide its MAC address while receiving the same IP address from the router.

The final option on this tab – MTU – is usually best left set to automatic, but on the rare occasions when a specific figure such as 1425 or 1500 is required, you can enter it here.

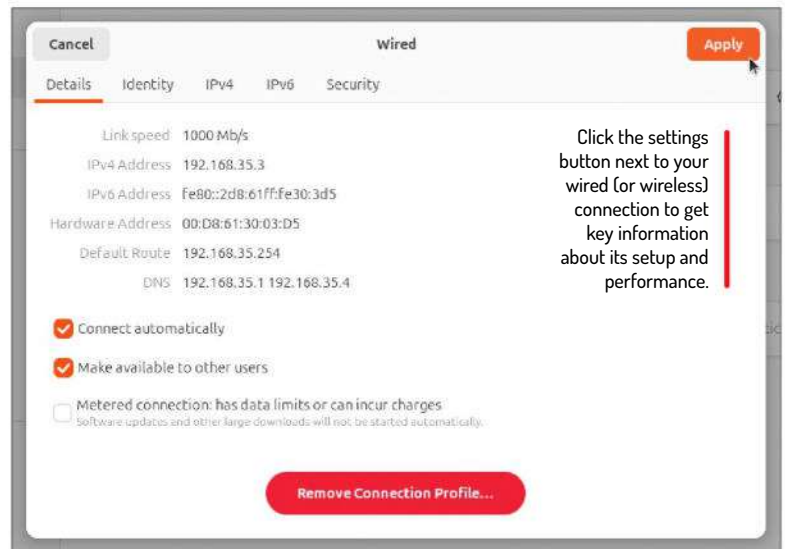
Inside IPv4

The IPv4 tab is where you can change your PC's IP address if needed. By default it's set to Automatic (DHCP), which means your router assigns your computer an IP address from its pool of reserved addresses. For most people, this is absolutely fine, but there may be occasions when your PC (such as a server) requires a permanent static address that ensures it can always be found by other devices on the network.

To change this, select Manual and fill in the Address (192.168.x.y where 'x' matches that of your router, and 'y' is a number between 0 and 255 that's not being used by another device), Netmask (24) and Gateway (the IP address of your router) fields. Ubuntu supports multiple addresses, but in most cases you only need the one.

Other options for the IPv4 Method include Link-Local Only, which basically allows two or more devices to connect with each other without requiring a router. Although there's no configuration involved, it's worth noting that your PC will be assigned an IPv4 address in the region of 169.254.0.x, where 'x' is a unique digit between 0 and 255. Finally, Shared To Other Computers is designed to connect two devices directly together via a single Ethernet cable plugged directly into each.

Below the IPv4 Method is DNS, where you can bypass your router's DNS server settings with your own, which will apply to that PC only. Favourite alternatives include 1.1.1.1 and 1.0.0.1 (Cloudflare), or



8.8.8.8 and 8.4.4.8 (Google). The final option – Routes – is best left set to automatic.

The IPv6 tab works in a similar way, but few people will need to use it, so you can skip this section. The same is true of Security, which is typically only required when logging into a large network, like an organisation or university.

Expert wired options

If you've signed up to a VPN service, in most cases you'll simply follow the instructions given to gain access to it – some providers create VPN profiles when you set up their client or provide you with a file you can import when clicking the + button next to VPN.

There may be times when you need to set up a manual connection – out of the box, Ubuntu supports OpenVPN, PPTP and WireGuard protocols. Simply select your chosen protocol and then fill in the details required – typically you need to know the Gateway (a public IP address or domain) and have access to a username/password combination or private key to gain access to the network.

The final option – Proxy – offers a simple on/off switch and choice of automatic or manual

QUICK TIP

Your public IPv4 address is what is allocated to your router by your internet provider, and usually changes over time – typically when you reboot your modem. You don't need to know this address for day-to-day use, but you can find out what it is by visiting www.whatismyip.com.

» STREAM MEDIA FROM YOUR PC

Turn your PC into a simple media server by navigating to Settings > Sharing and clicking Media Sharing. Flick the main Media Sharing switch to on and compatible media in your **Music**, **Videos** and **Pictures** folders will be accessible to any streaming device capable of connecting to a DLNA or UPNP server, such as the Roku Media Player app or your smart TV's own video player.

They'll see your PC listed as an accessible device, and then they'll be able to browse by folder to the content they wish to view or listen to – so long as you use universal media formats such as MP3 (music), JPEG (photos) or MP4/MKV (video), you should have no problems.

You can choose what folders to share, enabling you to exclude any personal folders or add others if media is stored elsewhere (say on an external drive). If you're connected to multiple networks, you can also restrict the server to only working on a specified network. The network names are a little cryptic but trial and error should see you able to pick and choose what networks your server functions on.



QUICK TIP

You can also get lots of useful information about your network by logging on to your router's administration utility. Open a web browser and navigate to its IP address – consult the manufacturer's support page for the username and password needed to log into it (we recommend changing this for security reasons).

configuration. Again, this isn't normally required in a home user environment.

Wi-Fi settings

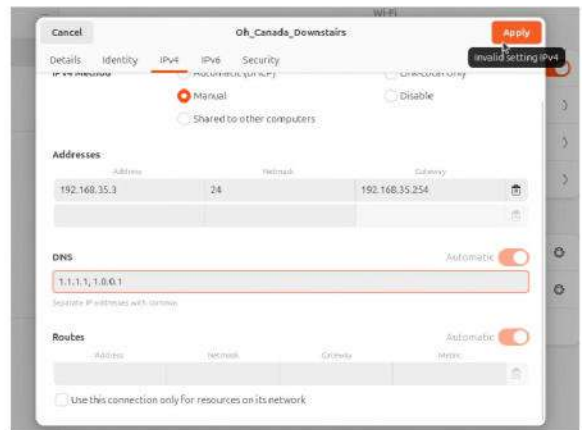
The Wi-Fi section of Settings superficially looks a lot different from the Network section. At the top is a simple toggle for switching it on and off – the same as clicking the orange button in the menu bar pop-up. Below this you'll see three sections – click > next to each to expand them in turn. Below this is a Visible Networks list that shows you any networks within range of your PC, including the one you're currently connected to. Click another hotspot in the list to switch to it – you'll need to enter its password to complete the connection if you've not previously used it.

Saved Networks contains a list of all the networks to which you have previously connected. Click the Settings button next to a connection and you'll see a multi-tabbed window similar to that for wired connections appear.

This works in pretty much the same way, but there are some differences – the Details tab provides useful information about the strength, speed and security of the connection, while Security is of more interest to all users as it enables you to switch security protocol (say from WPA2 to WPA3 if supported) and change the password if necessary. Click Apply when done.

The Saved Networks window also enables you to delete redundant connections to keep things tidy, plus you'll also see a barcode icon – click this to bring up a QR code with your network name and password, which others can use to connect to your network.

Below Saved Networks are two more options. The first is Connect To Hidden Network. This can be used to connect to a network that isn't broadcasting its SSID (and so doesn't appear when scanning for networks) –



If your PC needs to have a permanent – or static – IP address, switch to the IPv4 tab to configure it.

you need to know the SSID, of course, plus the security protocol used to encrypt the network.

PC hotspots

The final option enables you to convert your Wi-Fi adaptor into a hotspot for others to connect to. You need an alternative connection to your network (typically Ethernet, but maybe a second Wi-Fi adaptor if you have one) for this to work. Simply provide a network name (basically an SSID to identify it to others) and review the randomly generated password, which you can change (the security is automatically set to WPA, which should work with all modern devices). Click Turn On to enable the connection, at which point the list of Visible Networks at the bottom of the screen is replaced by a QR code that mobile devices can use to connect to your new hotspot. You'll also see a prominent red Turn Off Hotspot button for when you need to remove the connection for whatever reason.

Network resources

Once you've established your network connection is working correctly, you'll want to start making good use of network resources like shared folders and printers. We've touched on the former in previous *Linux Basics* tutorials – most recently last issue when we revealed how to install Samba to provide you with the ability to share folders from your own PC with other networked machines including those running Windows or Mac OS.

When it comes to connecting to shared folders on other devices, open Files and click the Other Locations entry, then scroll down to the Networks section to view visible devices on your network. Click a device and – if it's password-protected – you're prompted to enter a username, domain and password. The username and password need to match a user account set up on the remote device, while Domain – set to WORKGROUP by default – can usually be left as it is. Enter your details and choose whether to have the login details forgotten immediately, remembered until you log out of Ubuntu (the default setting, which means you would need to enter them again each time you restart your PC), or remembered for ever. Make your choice and then click on Connect.

If successful, you should see a list of folders shared by that device, enabling you to browse and access them like any other folder. If you're unable to copy or

» TROUBLESHOOT YOUR WI-FI

Problems connecting to your Wi-Fi network? First, verify your network adaptor has been detected by going to Settings and checking for a Wi-Fi entry. If it's there, select it and check to see whether you're able to see any Wi-Fi networks to connect to.

If it's not there, check to see if it's been detected by Ubuntu. Press Ctrl+T to open a terminal window, then type the following command:

```
$ lshw -C network
```

If your hardware has been detected, it'll be listed here. Try unloading and reloading your Wi-Fi driver. First, identify the driver with the following:

```
$ lspci -k | grep -i network -A 3
```

Then issue the following commands, replacing 'iwlwifi' with whatever is listed under Kernel Driver In Use:

```
$ sudo modprobe -r iwlwifi && sudo modprobe iwlwifi
```

```
$ sudo systemctl restart NetworkManager.service
```

This should hopefully bring your Wi-Fi device back to life – verify it has with the following command:

```
$ nmcli device
```

If the Wi-Fi adaptor isn't present, check the dongle (or PCIe card) is plugged firmly into your PC, then visit the manufacturer's website to see if there are any known issues with Linux or drivers you need to download. Alternatively, visit <https://github.com/morrownr/USB-WiFi/> for more help and advice identifying your USB Wi-Fi adaptor and getting it to work in Linux.

edit files on the remote server, check the permissions for the user you've used to connect on the remote machine – they may only be set to read-only.

Troubleshoot shared folders

Some network devices may refuse to appear in the list – this doesn't mean they're not visible or accessible, just that for some reason or other they're not showing up. In these cases, you can make use of the Enter Server Address box to connect directly to them by network name or IP address.

Before typing the address, click the ? button to the right of the input field to bring up a list of supported protocols and how you go about connecting to them. For example, to use Samba, you need to precede the device's address with **smb://** like so:

smb://n100-server

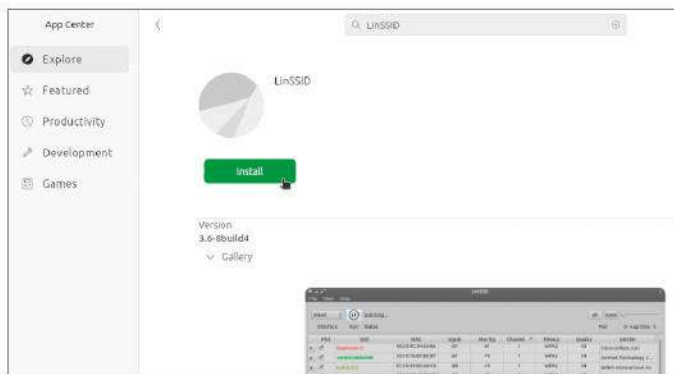
If that fails to connect, try the device's IP address – **smb://192.168.1.2** for example.

Other network resources

As in Windows and Mac OS, Ubuntu makes it incredibly easy to connect to network printers. In fact, you'll find in many cases they're quietly set up in the background – when you come to print, you'll find the printer is already set up and ready to use. Verify which printers are available via Settings > Printers, where you'll see an Add Printer button that scans the network for connected printers. If yours doesn't appear, enter its IP address into the search box to see if it shows up.

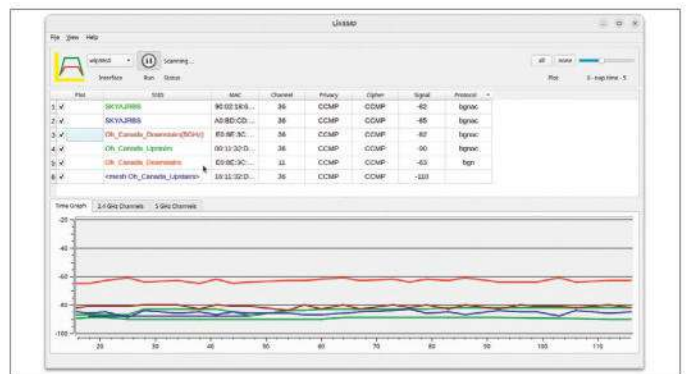
Still no luck? Either dig out your copy of **LXF309** for our in-depth tutorial on setting up printers in Linux, or wait for next issue, when we'll look at printers as part of a wider look at setting up hardware drivers. **LXF**

IMPROVE WI-FI PERFORMANCE



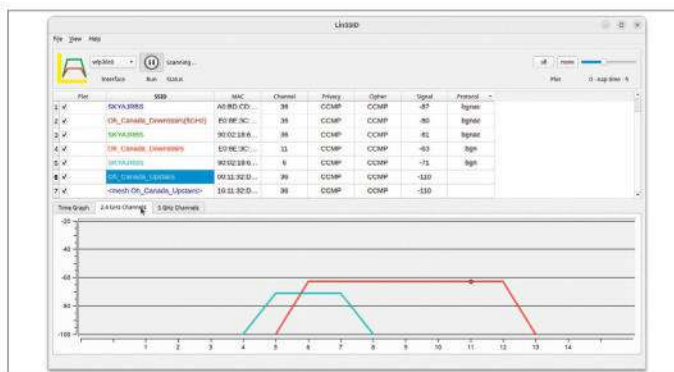
1 Install Wi-Fi scanner

Open **App Center**, type **LinSSD** and then click **LinSSD** from the drop-down results list under **Debian Packages**. Click the green **Install** button followed by your user password to install the program. Once complete, close **App Center** and open **LinSSD** from the **Launcher** (it'll appear at the end of the list). Enter your password again when prompted.



2 Perform a network scan

When the main screen appears, click the grey play button near the top and **LinSSD** will begin monitoring the performance of any Wi-Fi networks that are within range. Each network is displayed in a list, (multiple entries indicate both 2.4GHz and 5GHz channels) along with a graph showing their relative signal strengths over time.



3 Check for congestion

Switch to the 2.4GHz tab. If there are lots of overlapping networks sharing the same channel, your Wi-Fi performance will suffer. The dot on the graph shows the current channel your Wi-Fi adaptor is currently broadcasting on. It's a similar picture with the 5GHz channel.



4 Switch channels

Log into your router's configuration utility (typically by typing its local IP address into your browser), then locate the Wi-Fi settings page for each channel. Try changing the channel to one that's not in use or closely overlapping other devices – for example, on the 2.4GHz channel, try channels 1, 5 or 11.

» GET TO GRIPS WITH MORE LINUX! Subscribe now at <http://bit.ly/LinuxFormat>

Advanced PDF editing and tricks in OnlyOffice

PDF is an extremely popular format, touching our workflow every day. **Kseniya Fedoruk** shares tips to get the most out of dealing with them.



OUR EXPERT

Kseniya Fedoruk has been a Kubuntu and open source enthusiast for years, a contributor on GitHub, and is a document expert at OnlyOffice.

PDF documents are frequently used in professional settings due to their wide range of applications, dependability and compatibility. They play a crucial role in maintaining consistent document formatting regardless of the device or platform being used, guaranteeing the content is displayed accurately. PDFs are instrumental in sharing critical documents, designing forms and securely disseminating information in a polished manner. And this tutorial aims to offer key strategies to optimise your PDF workflows for the best results, using the OnlyOffice suite.

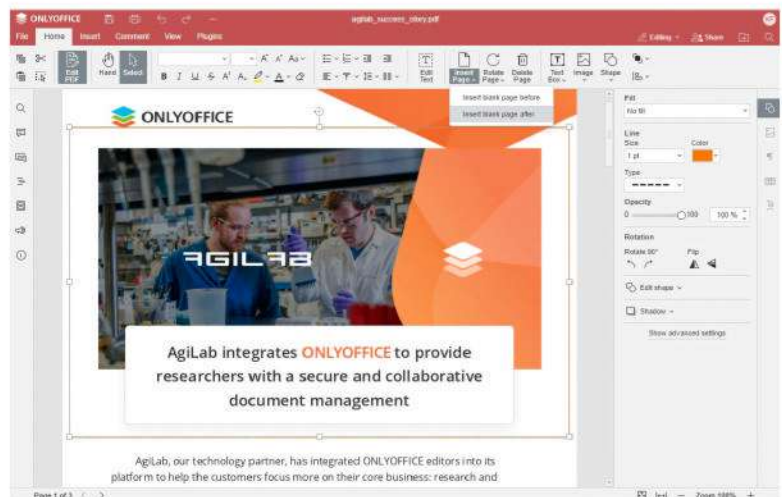
You can easily open and browse through PDF files, and this functionality comes in handy for when you want to quickly go through documents or retrieve information from PDFs without the hassle of converting them to a different format.

The built-in PDF viewer allows for zooming in (also fitting to page and fitting to width), flipping through pages, searching for specific text, as well as printing. This makes it convenient for you to locate the desired information and effortlessly review documents.

If necessary, switch to the View tab to use the Headings option, change the interface theme, or adjust the display of the interface elements (toolbar, left panel and status bar).

Annotating PDFs

An effective way to offer feedback or emphasise key details is by including annotations in PDF files. The



There is a wide range of tools available for editing a PDF in OnlyOffice.

annotation tools available in *OnlyOffice Docs* – commenting, highlighting, underlining and striking through text, as well as freehand drawings – promote teamwork when you need to review and discuss documents.

The annotation mode comes with a mini toolbar to make working with annotations easier. This enables you to view all annotations on the text, add new ones, and remove any that are no longer needed. It is also possible to remove annotations using the context menu, which helps streamline the review process and maintain a well-organised document.

Use the option to switch between modes in order to either view and annotate or edit a PDF file. This versatility enables you to customise your document interaction to suit your specific requirements.

The Viewing mode enables you to read and annotate a PDF file, while the Editing mode provides

QUICK TIP

Ever since the release of version 8.1, *OnlyOffice Docs* has introduced a native PDF editor that simplifies and streamlines the process of working with PDFs. It is accessible in both the online version and the free desktop app for Linux.



Here you can see the View tab in the OnlyOffice PDF editor.

access to a wide range of editing tools for making extensive changes to the document. To switch between the modes, use the top switcher or the Edit PDF button.

Editing PDFs

You can also directly modify the PDF file structure and contents. With the *OnlyOffice* PDF editor, it's possible to edit and format text, add, rotate or delete pages, insert and customise various objects (tables, shapes, text boxes, images, TextArt and so on), and insert hyperlinks.

It is also possible to generate interactive forms with fillable fields. This functionality is vital for designing applications, surveys and various documents that necessitate user interaction.

You can provide a variety of options for collecting the required information swiftly and accurately by adding various form field types: text field, combo box, drop-down, checkbox, radio button, image, email address, phone number, date and time, postcode, credit card details, or complex fields.

Once added, you can move form fields to another place in the document, as well as adjust field properties, which you can see in the right-hand toolbar – for example, you can limit the number of characters, allow text entry on multiple lines, divide the field into cells for each character, make a field obligatory, manage roles, and so on.

If you want to save yourself time, you can search for ready-made PDF form templates in the free *OnlyOffice* library: <http://oforms.onlyoffice.com>.

With *OnlyOffice*'s interactive form field support, filling out and saving completed forms to a PDF file is simple. This feature eliminates the manual process of printing and filling out forms, ultimately saving time and reducing paper usage.

Use the View Form option to fill out the fields. Once the form is filled out, you can save the document and electronically share it, guaranteeing that the information is transmitted accurately and securely.

Protecting PDFs

It is essential to safeguard sensitive information by using password protection for PDFs. This measure guarantees that only approved users are able to view or modify your files, thereby upholding confidentiality and preserving the integrity of the documents.

By setting a password, you add an extra layer of security to prevent unauthorised access to the document. Additionally, applying permissions enables you to manage what actions users can perform on the PDF, such as filling out forms, tracking changes and leaving comments.

Converting files

With *OnlyOffice Docs*, you have the option to convert your PDF files into various formats such as DOCX, ODT, EPUB, FB2, JPG, PNG and more. This enables you to make significant edits to your documents with ease.

Furthermore, any file generated in *OnlyOffice*'s document, spreadsheet or presentation editor can be



exported as a PDF, allowing for consistent formatting across various platforms.

How to test PDF Editor

To test the PDF editor, you can quickly install *OnlyOffice Docs* or DocSpace on your own server. The recommended option is to use *Docker*. For *OnlyOffice Docs*, the *Docker* command is the following:

```
$ sudo docker run -i -t -d -p 80:80 --restart=always -e
JWT_SECRET=my_jwt_secret onlyoffice/
documentserver
```

Detailed steps on installation of *OnlyOffice Docs* are covered in the official GitHub repo: <http://github.com/ONLYOFFICE/Docker-DocumentServer/>. For *OnlyOffice DocSpace Server*, the main commands are:

```
$ wget http://download.onlyoffice.com/docspace/
docspace-install.sh
```

```
$ bash docspace-install.sh docker
```

Find the complete instructions in **LXF314**, or create a free cloud account on the official website.

You can also try the PDF editor in the desktop environment (*OnlyOffice Desktop Editors*). Use the installation option via Snap package, for example, which is available for lots of Linux distros:

```
$ snap install onlyoffice-desktopeditors LXF
```

OnlyOffice enables you to build complex PDF forms.

QUICK TIP

Do you have a PDF document that needs to be converted? Use the free online converter to instantly change its format without needing to open the actual editor: <http://onlyoffice.com/pdf-converter.aspx>.

» ONLYOFFICE & DOCSpace

OnlyOffice Docs is an open source office suite that comprises online viewers and collaborative editors for text documents, spreadsheets, presentations, forms and PDFs. You can use it within *OnlyOffice DocSpace* or other cloud platforms, such as Confluence, Alfresco, Moodle, WordPress, Odoo, Jira, Redmine, Nextcloud and so on, as well as embed the editors into your own service/app and deliver them to your end users.

OnlyOffice DocSpace is a room-based environment for online document collaboration with colleagues, teammates, customers, partners and so on. The platform allows organising a clear file structure depending on individual needs or project goals. Flexible access permissions and user roles allow fine-tuning the access to the whole space or separate rooms.

» **IMPROVE YOUR LINUX SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

BACK ISSUES » MISSED ONE?

ISSUE 319 September 2024

Product code:
LXFDB0319



In the magazine

Get a flavour of what the latest version of Linux Mint has to offer, and dive inside Linux to discover the audio stack and learn what PipeWire has to do with it. Plus, explore the Linux filesystem, find out about NixOS, expand your storage with RAID, control multiple AI models, and record games. And we haven't even mentioned our packed reviews, Raspberry Pi and news sections...

ISSUE 318 Summer 2024

Product code:
LXFDB0318



In the magazine

Find out how to boost your VPN privacy and set up a free VPN, and continue keeping your secrets safe with our *Roundup* of security-conscious chat services. Plus, try our tutorials on everything from tweaking your install settings to ray tracing on your Steam Deck, and plenty in between. And don't forget to read our packed news, reviews and Raspberry Pi sections to stay up to date with open source.

ISSUE 317 August 2024

Product code:
LXFDB0317



In the magazine

Don't consign your old machines to the scrap heap – they can still be put to work with Linux. And join us as we dive deeper into Linux to explore virtualisation, containerisation and virtual filesystems. We also have tutorials on managing software and files, emulating VDUs, creating old-school pixel art, and more. Plus we have our usual news, views and in-depth reviews.

ISSUE 316 July 2024

Product code:
LXFDB0316



In the magazine

Transform your desktop with hot KDE Plasma, and join us as we start our journey inside Linux to discover how it works. Plus, choose the best text editor for your writing projects, find out about the phone firm making an environmental impact, build your own weather machine, get back to basics with the Linux terminal, keep your video calls private, and enjoy a wealth of reviews, news and views.

ISSUE 315 June 2024

Product code:
LXFDB0315



In the magazine

Discover how to upgrade to Ubuntu 24.04 LTS and join us on a journey through the world's most popular distro's 20-year history. And don't miss our *Roundup* of hacker distros, an in-depth look at the impending Epochalypse, detailed tutorials on live-streaming audio via the terminal, setting up Gnome, editing images in *Krita*, and archiving C64 tapes, and plenty more besides.

ISSUE 314 May 2024

Product code:
LXFDB0314



In the magazine

Join us as we explore Linux's infiltration of Windows, and discover how you can also get Windows apps running in Linux as we uncork the latest *Wine* release. And let's not forget a *Roundup* of time trackers, tutorials on vulnerability audits, setting up a home ebook server, streaming games, and more, along with distro reviews, news, and packed Pi and *Adminsteria* sections.

To order, visit **www.magazinesdirect.com**

Select **Single Issues** from the tab menu, then select **Linux Format**.

Or call the back issues hotline on **0330 333 1113**
or **+44 (0)330 333 1113** for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

USA? EU? THE MOON?

UK readers
turn to
p16

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you.



The image shows three copies of the Linux Format magazine. One is a print copy on the left, one is on a tablet in the center, and one is on a smartphone on the right. All three displays show the same cover: 'FRESH MINT 22' featuring a penguin on a laptop. The cover includes headlines like 'Update your PC to the slickest Ubuntu spin', 'New apps • Classic desktop • Matrix chat', and 'Secure Flatpak • The latest 6.8 kernel'. It also mentions '100 pages of Linux tricks, tips & more!' and 'PLUS: HOW TO' with sub-headings like 'Code a fun Wordle-style game', 'Try the cutting-edge NixOS', and 'Run the classic Symbian OS from Psion'. At the bottom, there are sections for 'INSIDE PIPEWIRE', 'STEAM STREAMS', and 'C64 ASSEMBLER'.

2 GREAT WAYS TO SUBSCRIBE

Print + NEW digital access or digital-only!

» USA
From \$135.49
For 13 issues

» REST OF THE WORLD
From \$135.49
For 13 issues

» EUROPE
From €119.49
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit www.magazinesdirect.com/linux-format

Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm GMT

Or 1-844-779-2822

Lines open Monday-Friday 8.30am-5pm EST

*We don't actually deliver to the Moon. Yet.

RETRO EMULATION

Emulate the classic Psion PDA range

Les Pounder travels back to the 1990s, when Filofax became digital, and your phone was dumb and the size of a brick.



**OUR
EXPERT**

Les Pounder is associate editor at Tom's Hardware and a freelance maker. He blogs about hacks and makes at <http://bigles/>.

QUICK TIP

The code used in this tutorial, along with extra images, can be found in our GitHub repository: <https://github.com/lesp/LXF-Psion-Emulation/archive/refs/heads/main.zip>

High school in the '90s is where we saw our first Psion PDA (personal digital assistant) that we knew was a Psion. A classmate brought in a Psion 3a with a series of solid-state disks (yes, SSDs!) that had games and programs on. We forgot all about art theory and just delved into the wondrous world of the Psion. We never truly owned a Psion – the nearest we came was owning a Psion 5mx for two weeks before the flat flex display cable broke – but we loved the Psion hardware, which brought a decent keyboard to the PDA form factor.

Psion began life in 1980 as a software house with close ties to Sinclair Research (which explains why we saw so many Psion games on the Sinclair machines). The name Psion, an acronym, actually stands for Potter Scientific Instruments. The ON is due to PSI already being an acronym, and company founder David Potter claims that it also stands for Or Nothing. Making Psion mean Potter Scientific Instruments Or Nothing, a testament to Potter's determination and ambition.

Psion's hardware story starts with the Psion Organiser in 1984. Looking like a chunky calculator, the Psion Organiser I had an unusual alphabetical keyboard layout and a single-line LCD screen. The screen was changed to a two-line model with the Organiser II, but the keyboard remained the same. These machines were loved in industry, including retail, with UK chain Marks and Spencer using custom branded models in



The Psion series has a classic design aesthetic that is still coveted to this day. The clamshell protects the screen and hides a stunning keyboard.

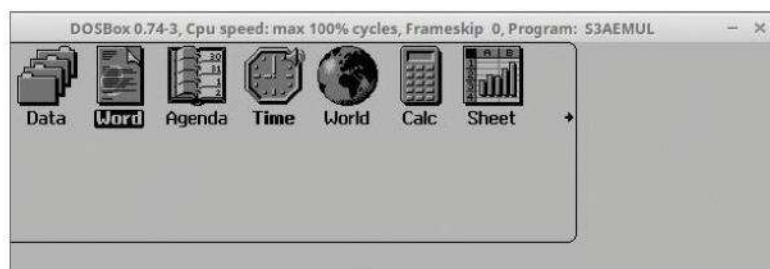
its stores. Even the UK government's welfare system used them for benefit calculations.

These early devices paved the way for the clamshell Psion Series 3 machines. Looking like a PDA, these devices were full-blown computers with their own programming language (OPL, introduced with the earlier Psion organisers) and a plethora of applications accessible via on-screen icons.

The Psion 3 series spawned the 3a, 3c and the 3mx. Each model improved on the original design with a bigger screen, faster CPU and more RAM. Each of these models shared the same clamshell design and a decent QWERTY keyboard, but it was the Psion 5 series that introduced the best keyboard of all.

The Psion 5 series was the last hurrah of the clamshell form factor and – wow! – what a send-off. With a sliding keyboard that felt great to type on, the Series 5 and the 5mx were powerful tools. Processor speeds were bumped up (to 36.864MHz) and RAM was increased, starting with 4MB on the Series 5 and reaching 32MB with the 5mx Pro. The screen size saw a slight boost, but it is the resolution that saw the biggest increase. We now had a 5.6 inch 640x240-pixel screen, which proved to be usable while commuting and still ran from two AA batteries.

The Psion range ended with the Psion netBook. Yes, Psion made a netbook before everyone else did in the late 2000s. Running the Symbian EPOC Release 5, this



The Psion Series 3 menu brings back a lot of memories. A simple, clear user interface that gets the job done.

device proudly carried the Psion legacy into the 2000s. Boasting a 7.7 inch, 640x480 256-colour LCD, 32MB of RAM and a 191.7MHz CPU, this was a much larger device. Measuring 23.5x18.2x3.7cm, this Psion was no longer a pocket computer. It did have a claimed 12 hours of power via an internal Li-ion battery. We had the opportunity to use one once at Blackpool Linux User Group, and it was a great experience, but we yearned for the pocketable form factor of the Series 3 and 5.

We digress. Here is where we leave the history lesson and instead learn first-hand about the original of the Psion range, the Psion 3a.

Emulating a Psion

Psion emulation is a bit of a grey area. First, there is the ROMs situation. Any ROMs for the Psion, just like other retro computers or consoles, are legally owned by the copyright holder. This doesn't stop people from downloading ROMs, but it is an ethical exercise for the user. We won't tell you where to get ROMs; that is left to your conscience. Secondly, the pool of emulators is a little shallow. So shallow, in fact, that we found the same Psion 3a emulator that we used on our 486 PC back in high school.

Let's start the emulation journey by installing *DOSBox* on our Xubuntu 24.04 test system. Open a terminal and first update the software repositories, then install *DOSBox*:

```
$ sudo apt update
$ sudo apt install dosbox
```

In a browser, download the Psion SIBO emulator via this link – www.primrosebank.net/computers/pda/psion3a/S3AEMULE.zip – and extract the contents of the archive into a folder called **psion3a** within the **.dosbox** folder inside of your **home** directory. For example, our setup saw **psion3a** saved to **/home/les/.dosbox**. The **.dosbox** folder is a hidden folder, which can be viewed in your file manager by going to View > Show Hidden Files (or Ctrl+H). Open the **psion3a** folder and create three more folders called **a**, **b** and **m**. A and B are the drive slots located on the left and right sides of the Psion, and these are typically filled with solid-state disks. M is the internal drive letter for the Psion machines. Now go back to the **.dosbox** folder and you will see a file called **dosbox-0.74-3.conf** (or similar). Open this file in a text editor and scroll down to the bottom.

This file controls how *DOSBox* behaves, and we need to use it to mount two virtual drives for our virtual Psion 3a. The first mount is for our emulator. We'll create a drive called **p** and point *DOSBox* to the **psion3a** folder:

```
$ mount p ./psion3a/
```

Next we'll create the lettered drives (**a**, **b** and **m**) for our Psion 3a. It is important to create all of these drives, otherwise the applications might throw out error messages:

```
$ mount a ./psion3a/a/
$ mount b ./psion3a/b/
$ mount m ./psion3a/m/
```



Save the file and close the editor, then open a terminal and run *DOSBox*:

```
$ dosbox
```

Because we made those config changes, *DOSBox* will load our Psion 3a emulator as drive **P** and its internal drive is **M**. But we need to navigate to drive **P** to load the emulator, which we can do using these *DOSBox* commands:

```
P:
```

Now run the emulator:

```
S3AEMULEXE
```

After a few seconds, we're greeted with one of the most welcoming and usable portable operating systems that we have ever used.

Psion's operating system - EPOC

EPOC, coming from the word epoch, the beginning of an era, was the OS for all models of Psion from 3 onwards. Using a well-realised graphical user interface that followed the WIMP interaction style (windows, icons, menus, pointer) but also using touch inputs for common functions and applications, EPOC was a joy to use on the go.

The Psion 3 introduced this new interface. Previous models had a single or dual-line LCD screen but the Psion 3, released in 1991, had a glorious 4.3-inch 240x80-pixel display that displayed your apps and files. Under the screen was a series of inputs for commonly used applications. These were moved up to the screen in the Psion 5 series.

Our first GUI app using OPL is a classic question to the user. It's not fancy but it explains how to write code for the GUI.

QUICK TIP

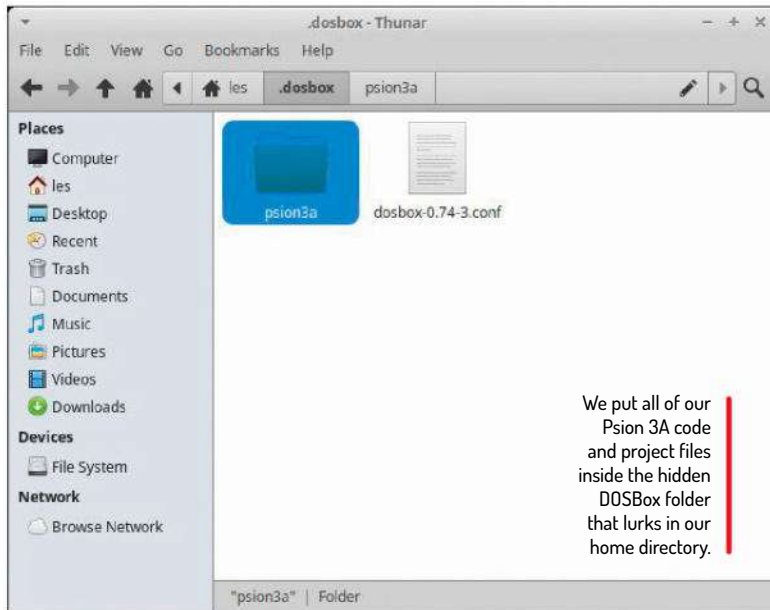
OPL calls files and programs modules, and all of these terms are used interchangeably to identify the code used in an OPL project. Modules continue at least one procedure (PROC) that makes up the main body of code.

» 21ST CENTURY PSION ONLINE

A long time ago at a Barcamp Blackpool event, we met someone called Kian Ryan and talked about software development and other very business-like things. But it wasn't until many years later that we learned that Ryan was an avid retro enthusiast and lover of the Psion series. In late 2022, Ryan was working on a PiRS232 sidecar (<https://bit.ly/LXF320Psion>), a device that connects to the Psion 5mx's RS232 connection to become a PPP modem. The glue that made this possible was the Raspberry Pi, specifically the Zero W. This project covers many aspects of maker skills. Firstly, there are the electronics to make it work, power in the form of a LiPo cell, and a Pimoroni LiPo Amigo Pro, which handles charging the cell and powering the sidecar. Finally, there is a 3D printed case that contains all of the electronics in a lovely shell that won't scare people when you use this setup in a coffee shop.

Ryan's project produced two different use cases. The first is a portable Linux terminal, accessing Raspberry Pi OS, installed on the Raspberry Pi Zero W, and letting Ryan do some sysadmin tasks on the fly. The other use case was an internet gateway. Using the Psion 5mx's browser app, Ryan could go online and surf the web. Sort of.





QUICK TIP

Don't throw out that old Series 5 or 5mx because the screen is broken. Chances are it is the flat flex cable between the body and the screen. They can be repaired, for a price. Or you can do it yourself.

In our emulated Psion 3, we can navigate the EPOC operating system using the cursor keys and Enter to start an application. Additional keys are F1 to go back to the main EPOC OS homepage, while F9 opens a general menu for file operations and application-specific actions. F10 opens a help page, and Esc closes any open menus. Pressing F2 to F8 opens (or selects) an application. After you have been using the Psion emulator for a while, you will notice that application icons have additional strings of text beneath them. These are your files, created in that application. Their syntax is **<File name> [Drive saved to]**. Remember that our emulated Psion has three drives to save to: **A**, **B** and **M**.

The Psion series devices were notably used as PDAs and office tools. We could send emails using the Psion 5mx, even surf the internet, but did you know that you can make your own applications and tools using a built-in programming language?

Using the cursor keys, navigate to OPL and press Enter to start a simple project.

Psion OPL

Open Programming Language (OPL) was created by Psion in 1984 and used on EPOC and later Symbian systems. OPL is an interpreted language that is similar to Basic, but it is specific to Psion and Symbian devices. We'll be using OPL3 on a Psion 3 emulator; Psion 5 and 5mx machines used OPL5, which saw a number of refinements.

OPL needs to be translated before it can be run. Think of it like Basic; the human readable code needs to be converted into something machine readable.

The process for creating an application or script in OPL is to type in the code, translate it into something that the Psion can run, then run the application.

Press F9 and from the File menu select New File. Call the file **name** and save it to disk **A**. Press Enter. We are now dropped into the editor, with two lines of code: **PROC** and **ENDP** – **PROC** is the start of our procedure, our main body of code, while **ENDP** is the end of the procedure.

Give your procedure a name – for simplicity, we chose **main**:

PROC main:

Press Space twice and create a local variable called **name**. The 20 in parentheses refers to how long the string can be:

LOCAL name\$(20)

Create another local variable called **i** – this will store an integer used in a loop:

LOCAL i%

Now we can give an instruction to the user via a **PRINT** statement:

PRINT "Enter your name: "

Using the **INPUT** keyword, we capture the user's response and store it to the **name** variable:

INPUT name\$

Using a **WHILE** that counts up from 0 to 9, we'll print the user's response, pausing for 0.1 seconds between each repetition. Then we increment the value of **i**, so that it increases by one each time the loop iterates. Note that the three lines of code inside the **WHILE** loop are indented two spaces to indicate that they are part of the loop:

```
WHILE i%<10
  PRINT "Hello, "; name$
  PAUSE 10
  i%=i%+1
```

Finally we end the **WHILE** loop, and use **GET** to wait for the user to press any key, exiting the project code.

ENDP signifies the end of our project code:

```
ENDWH
GET
ENDP
```

Press F9 and select File > Save. Press F9 and select PROG > Translate. Select Y to run the program. Enter your name and watch as the code prints it 10 times. Press any key to exit when the code stops.

Using OPL, we can also create GUI applications that use the EPOC OS dialog boxes. We'll generate a dialog box that asks a question and then uses the response with an **if** conditional test to generate a message to the user via another dialog box.

Create a new file (F9 and from the File menu select New File) and call the application **LXF** and save it to drive **A**.

Call the procedure **lxf**:

PROC lxf:

Next we create the dialog that will ask the eternal question: 'Is Linux Format the best Linux magazine?' The title of the dialog is **dINIT**, and the text is **dTEXT**. The user's response is captured using **dbUTTONS**, which generates Yes and No.

dINIT "Linux Format"



```
dTEXT "" "Is Linux Format the best Linux magazine?"
dBUTTONS "Yes", %y, "No", %n
```

Using a conditional test, we check the user's input. If the input is Yes, we create a new dialog titled **Correct** and the text to reward the user. In this case, the Grail Knight's responses from *Indiana Jones and the Last Crusade*. We then render the **DIALOG** to the screen:

```
IF DIALOG=%y
dINIT "Correct!"
dTEXT "" "You have chosen wisely"
DIALOG
```

But what if the user answers that *Linux Format* is not the best Linux magazine? This will make us sad but we have to plan for this contingency. The **ELSE** condition triggers another dialog titled **Incorrect** and advises the user that they have chosen poorly. The **if** conditional test is then ended using **ENDIF**.

```
ELSE
dINIT "Incorrect"
dTEXT "" "You chose, poorly"
DIALOG
ENDIF
```

We then print a message to the screen advising the user to press any key to exit; the **GET** keyword captures any user keypresses and the **ENDP** keyword ends the procedure:

```
PRINT "Press a key to exit"
GET
ENDP
```

Press F9 and select File > Save. Press F9 again and select PROG > Translate. Select Y to run the program. A new dialog pops up, so answer the question and watch as another dialog appears to respond to your input. Press Enter to close the dialog and then any key to exit.

Software for the Psion

The Psion had plenty of games and applications to keep us amused. Many are freeware and we visited <http://sprightly.co.uk/psion3/classics.html> to enjoy a few of these classics. These games are designed for real Psion hardware, and as such there may be a few that don't work or run way too fast — just like our game of *Bricks*, an *Arkanoid/Breakout* clone.

Software installation is relatively straightforward; we'll use the *Bricks* freeware game as an example. Downloading the ZIP archive, we extracted the contents to our **M** drive (`/home/les/.dosbox/psion3a/m`). This creates a new directory called **APP** and a text file called **BRICKS.TXT**. Start your *DosBox* session, and load the Psion 3a emulator.

To install the game, press F9 and select Apps > Install, then press Enter. The install dialog asks for the filename, and this should be prepopulated with **Bricks.opa**. Disk is set to internal (**M**) and Position refers to where we want the app to appear in the apps list. Leave this as current and press Enter. The *Bricks* game is now installed and can be run by pressing Enter. Note that the emulator will

» I WANTED A PSION 5MX!

The elephant in the room is Psion 5mx emulation. To the best of our knowledge and our research, there is no emulator of the same ilk as the Psion 3A emulator that we used in this feature. There is an emulator, found at <https://wuffs.org/WindEmu/index.html/>. This online emulator is rather good, but it is limited when compared to the emulator we used. Any files we create are lost as soon as we refresh or close the page.

The online emulator runs fast and we can easily play with the built-in applications. From a coding aspect, OPL on the Series 5 is different from the Series 3, and while we could get our name game to work, our GUI example required significant rework due to a change in syntax. You'll also come up against the Psion 5 keyboard layout being absolutely faithful to the original hardware, and not your current keyboard. For all of the =, < and ; keys, we needed to use Alt codes. We can access the Psion's insert special characters option (Shift+Ctrl+C) and select some, but those ones were missing. Instead we used this reference for the codes: www.alt-codes.net.

If you want to try out the back-end to this online emulator, its creator, Ninji, has a website full of information on how it all works — <https://wuffs.org/blog/building-a-psion-emulator> — but we just couldn't get it to work on our test machine. All the source code can be found on GitHub at <https://github.com/Treeki/WindEmu/>.

forget that the application has been installed, and this step will need to be repeated.

Psion scene today

If our recent post on Mastodon says anything (<https://bit.ly/lxf320psion>), there is a lot of love for these machines. A search on eBay revealed that the Psion 5mx is prized. We saw an unboxed unit go for £80; boxed and complete, £200 is the average. Why do they go for so much? Because they are still functional and relevant. They can go online and do simple tasks, we can record information on the go (spreadsheets, notes, calendars) and still sync with modern equipment. There are hoops to jump through, but it can be done, even with our Linux machines.

Psion's legacy

Made for the business person on the go, Psions were the pinnacle of keyboard-based PDAs. The PalmPilot craze came and went, but it was the Psion form factor that captured the hearts of many. A rich ecosystem of software, still available, and hardware projects to augment their abilities make the Psion series alive and kicking. Just don't close the clamshell too hard — those screen cables are fragile; ask us how we know! **LXF**

QUICK TIP

In OPL, you don't have to indent your code as the interpreter is not whitespace-sensitive like Python. We do it in the tutorial as a way to clearly indicate if the code is inside a loop or conditional test.

Installing extra apps on the Psion 3A emulator is simple. Copy your files to a drive, then from the main menu, press F9 and go to Apps.



» **GET MORE OUTDATED TUTORIALS!** Subscribe now at <http://bit.ly/LinuxFormat>

ENCRYPTION

Reviving the WW2 Enigma machine

The Enigma machine was hugely sophisticated for its day. Today, all that clever mechanics can be replicated in software, as **Mike Bedford** reveals.



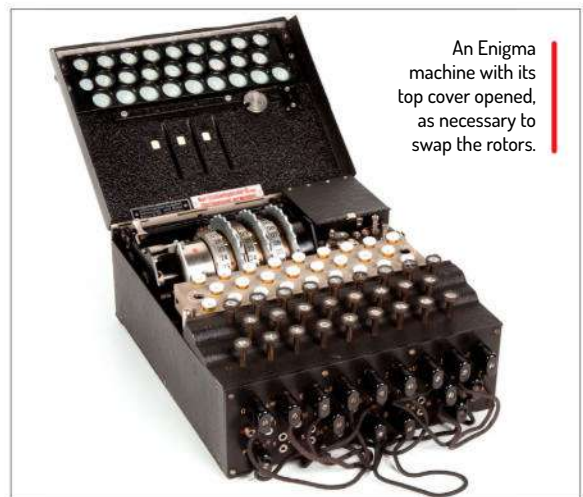
**OUR
EXPERT**

Mike Bedford was predated by the Enigma by quite some time, unlike several of the machines we've emulated, so seeing it in operation through emulation was quite an eye-opener.

The concept of the key length of a cryptographic system is familiar. A 128-bit key is commonly used today in a symmetrical cipher, because it's considered that a brute force attack – trying out all possible keys – is not feasible. Given that there are two to the power of 128 possible keys – that's 340,282,366,920,938,463,463,374,607,431,768,211,456 – this seems quite likely. Indeed, it's been said that if every person on Earth had a computer capable of checking a billion keys per second, all those machines working together would take billions of years to crack a message encrypted with a 128-bit cipher.

And so we come to the Enigma machine, the electromechanical contrivance used for encrypting and decrypting German military communications during WW2. We'll look at how the number was derived later, but for now let's say that the key length was around 77 bits. Admittedly, that gives 2,251,799,813,685,248 (around 2.25×10^{15}) times fewer keys than today's 128-bit ciphers, but we mustn't forget that computing equipment was almost non-existent in the 1940s. And to illustrate how keys substantially shorter than today's norm were previously adequate, a 56-bit key length was considered secure throughout the '70s. We might be tempted to assume, therefore, that Enigma would have been uncrackable in its heyday.

However, this ignores one very important fact. The security of a cipher doesn't depend only on its key



An Enigma machine with its top cover opened, as necessary to swap the rotors.

length. To give a trivial example, let's think about the mono substitution cipher in which each letter of the alphabet is transformed into another. The number of possible keys is 26 factorial, which equates to about 4.0×10^{26} , or around 88 bits. However, given a moderately long encrypted message, with a bit of patience you could decipher it by hand. The secret is that some letters are found in the English language more commonly than others. So, for example, knowing that E and T are the most commonly encountered, that

» CRACKING ENIGMA

Given the effective length of the key and the large number of operations before the machine returned to the initial state, it's incredible that Enigma messages were routinely deciphered at Bletchley Park. So, what was the secret of this remarkable achievement?

First, Enigma could never encrypt a letter as itself, and this was a flaw in its design, but this alone wouldn't have made it vulnerable to attack. The other aspect that helped codebreakers at

Bletchley Park was the way Enigma was used. The bottom line is that messages often had predictable words or phrases near the start, and usually included "Heil Hitler" near the end. If 10-letter groups towards the end of the message were analysed, any that had an H in the first or fifth position – bearing in mind that spaces weren't encrypted and transmitted – could not represent "Heil Hitler". Following similar logic, other groups could be discounted, thereby

providing a list of possible encrypted versions of that phrase – so-called cribs.

That's still a million miles from being able to crack the day's messages, and this is where the bombe came into its own. A bombe was effectively lots of motor-driven Enigmas working together in parallel, looking for possible settings that were consistent with the cribs. Still by no means an easy task, but hopefully you're starting to get just an inkling of the work at Bletchley Park.

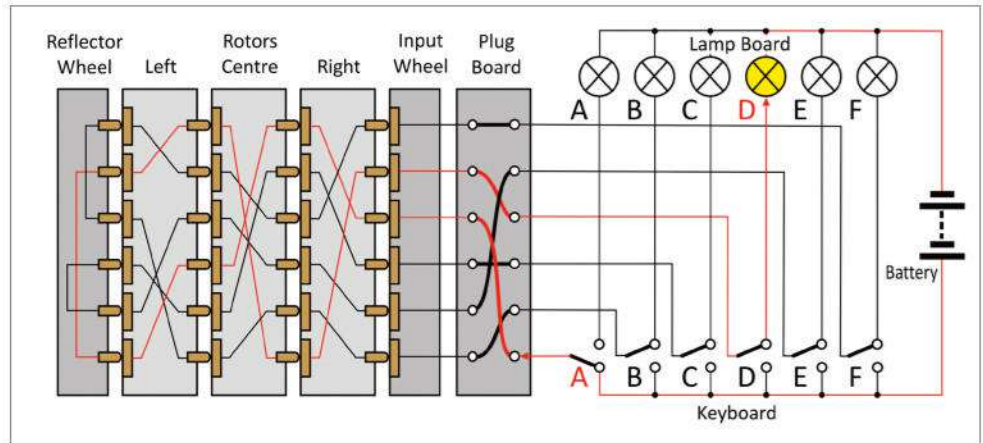
TH is by far the most common two-letter pair, and various other statistics, messages could be cracked in a reasonable timescale, even without computing assistance. As we're about to see, Enigma eliminated this massive weakness of the mono substitution cipher by changing the mapping between an unencrypted letter and the encrypted version after every letter. And the mapping only repeated every 16,900 letters.

Enigma in detail

An Enigma machine looked like an old-fashioned typewriter except, instead of a sheet of paper on to which letters were typed, it had a lamp board – an array of display lamps that represented the letters. To encrypt (or decrypt) a message, the letter was typed on the keyboard, and the lamp representing the encrypted (or decrypted) letter would illuminate. These letters would be transcribed by hand for radio transmission by Morse code or, in the case of a decrypted message, for delivery to the intended recipient. Between the keyboard and the lamp board, however, the electrical current took a convoluted path. Details of that tortuous path differed between different models of the Enigma, but the following description gives a feel for the operation of a three-rotor Enigma.

When a key was pressed, a current flowed to a plug board that had pairs of sockets for each letter. If a patch lead, which had a two-pin plug at each end, was not plugged into a particular pair of sockets, that letter would map to itself. However, if a patch lead was used to connect the pair of sockets for one letter to the pair of sockets for another, it would swap those two letters. So, for example, a lead between the A and the H socket pairs would convert A to H and H to A. Generally, up to 10 leads would be used in a particular setup.

From the plug board, the current made its way to the first of three rotors through which it would flow. These were wheels with 26 sets of contacts on one side and another 26 on the other side. Between these sets of contacts, wires internal to the rotors directed the current so each letter was converted to a different letter. In that respect, and also like the plug board, it performed a similar function to the mono substitution cipher we looked at earlier. In the case of Enigma, though, the machine as a whole didn't implement a mono substitution cipher and this takes us to a vital



aspect of the machine. Each time a key was pressed, the right-hand rotor moved on one position, so the mapping between unencrypted and encrypted letters changed. And to further improve security, once per revolution (and twice per revolution in the case of one of the wheels), the wheel immediately to the left – through which the current would pass next – also rotated. The upshot was that the Enigma machine only returned to the same state every 16,900 letters.

So, as we've seen, the current representing a letter passed through the plug board and three rotors, but we're not done yet. The current then passed to something called the reflector, which was rather like a rotor except that it didn't rotate. And as the name suggests, from the reflector the current was sent back through the three rotors in the opposite direction, then via the plug board to one of the display lamps.

We've seen that the alteration of the mapping between one letter and another changes after every letter, and only repeats after thousands of letters have been encrypted. This made Enigma immune to an attack by analysing the frequency of occurrence of letters. But what about the 77-bit effective key length that we quoted earlier? The answer lies in the fact that several aspects of Enigma could be changed before using it to encrypt messages. This setup was carried out periodically, and is often referred to as the daily setting, although it could have been more frequent.

We've already seen one variable feature – the arrangement of patch leads used in the plug board – but there was much more. First, although only three rotors were used at any one time, Enigma machines had several rotors, each with different internal wiring, and setting up the machine involved using the correct three in their designated positions. Next, the ring setting was selected, this being the relationship between the letters on a rotor and its internal wiring. The setting up process also involved selecting the starting point for each rotor. This final setting was chosen by the operator, it was different for every message, and was transmitted to the receiving party using the daily settings. To complete setup, for some models of Enigma, it was also necessary to select the required reflector from the ones provided. Multiplying together all these variables gives the effective key length and made a brute force attack infeasible.

We give some very brief information on how such an apparently uncrackable cryptographic system could possibly have been cracked back in the 1940s in the

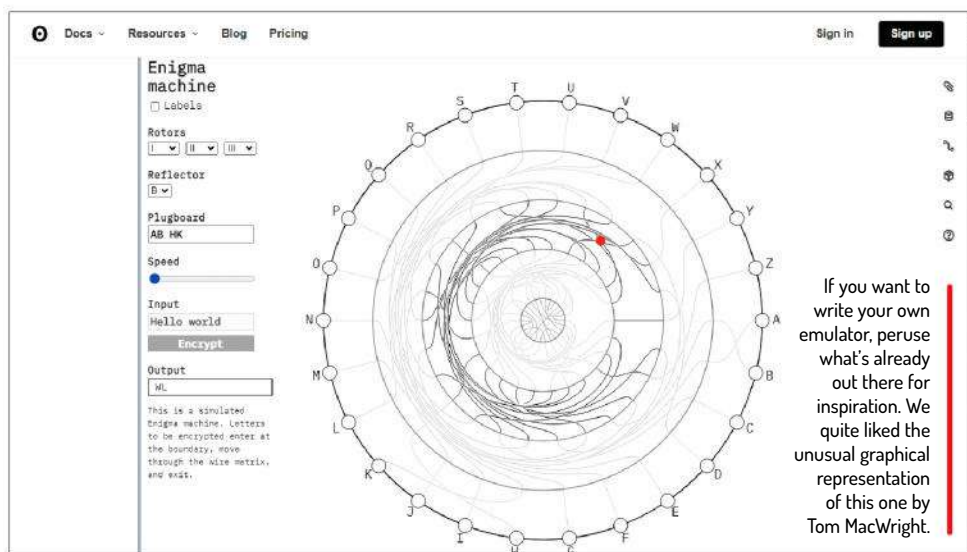
It might be simplified by having only six letters, but this wiring diagram illustrates the route taken between the keyboard and the lamp board to encrypt or decrypt a letter.

QUICK TIP

Getting hold of Enigma machines – both the pre-war commercial version and captured military versions – was considered key to the work at Bletchley Park. Things are very different today because it's assumed that, sooner or later, the details of any cryptographic method will become known. As such, current practice is that only the keys are kept secret.



A current passed from one rotor on the Enigma machine to another via spring-loaded pins on one to brass contacts on its neighbour.



the fact that the position of each rotor caused its left-hand neighbour to move on a position. Also, we looked at what we could almost refer to as a generic Enigma machine, but you might prefer to emulate one of the various models that existed, or perhaps allow one of several models to be selected.

The simplest approach is to write an emulator that takes an unencrypted message – or plain text, as cryptographers would call it – and displays the encrypted message, or ciphertext to cryptographers. However, you could go much further if you fancy more of a challenge. In particular, you could introduce a graphic display to show the electrical pathway that

QUICK TIP

A cryptographic device that requires the encrypted or decrypted text to be written down manually sounds incredibly primitive. And it did mean that a second operator was needed to achieve adequate speed. However, a printer was available for the Naval Enigma that transcribed the output on to a narrow strip of paper.

boxout (page 68). However, we will mention one thing here as it relates to the inner working of Enigma. Despite its internal complexity, it was a design aim that Enigma should be easy to use. So, the only aspect that required a degree of knowledge that wouldn't have been obvious was configuring the daily settings. After that, it was a simple matter of typing in the message, one letter at a time, and noting the encrypted equivalent as shown on the lamp board.

An important part of the simplicity of Enigma is that it wasn't necessary to do anything different for decrypting a message – in other words, there was no concept of an encrypting and a decrypting mode. This was possible because the process was reversible, so if A encrypted to H, then H would encrypt (or decrypt) to A. This, in turn, was achieved by the reflector. However, it also meant that a letter could never be encrypted on to itself. And it transpires that this was considered a design flaw that Alan Turing and his team of cryptanalysts at Bletchley Park were able to exploit.

Emulating Enigma

Over the years, we've looked at many emulators here in the pages of **LXF**, often of computers from a past generation. And in most cases, while ready-to-go emulators were readily available, writing one yourself would have been quite an undertaking. Emulating Enigma, on the other hand, is a much more accessible coding exercise. It would be wrong to consider it as a trivial job, though, and this middle ground might make it an interesting exercise. Having said that, you will need to read up on Enigma in more detail for information that we glossed over, such as, for example,

each letter takes as it progresses through the Enigma machine, and the changing positions of each of the rotors. You could even push the boat out by providing a photo-realistic representation. Each of these approaches has been taken, as we'll see as we take a look at some freely available Enigma emulators. You could try these out to give you some inspiration for writing your own emulators. Alternatively, if you want further ideas, other emulators abound, but we quite liked the one created by Tom MacWright at <https://observablehq.com/@tmcw/enigma-machine>, specifically because of its rather quirky yet visually attractive graphics. On the other hand, if you don't feel inspired to churn out your code, just use one or both of our featured emulators to get a better feel for this most famous of cryptographic machines.

Enigma the easy way

Our first Enigma emulator can be used simply for encrypting and decrypting messages, but that doesn't teach you much about the working of this machine. We chose to feature it, though, because it also allows you to see the signal paths that represented the stage-by-stage encryption of each letter between the keyboard and the lamp board. It was written by Amir El Bawab, it's open source, and you can find it at <https://github.com/amirbawab/Enigma-machine-simulator>. There are console and GUI versions; we used the GUI version.

There are three display formats, referred to as Text Box, Keyboard and Wires Connection. The Keyboard view is the closest to what it would have been like to use an Enigma machine, as it shows the keyboard and the lamp board. It's slightly more sophisticated than a true Enigma – which required each encrypted letter to be written down – because the plain text and the ciphertext strings are shown above the keyboard and lamp board respectively. However, you're limited to a single line of letters. So, if you want to encrypt a long message, you should use the Text Box display instead but, let's face it, you're not going to use an Enigma emulator for real-world applications. In the main, therefore, we suspect it'll be the Wires Connection display that you'll find most interesting.

As with the other two views, the top of the screen shows the settings for the rotors and reflector, while



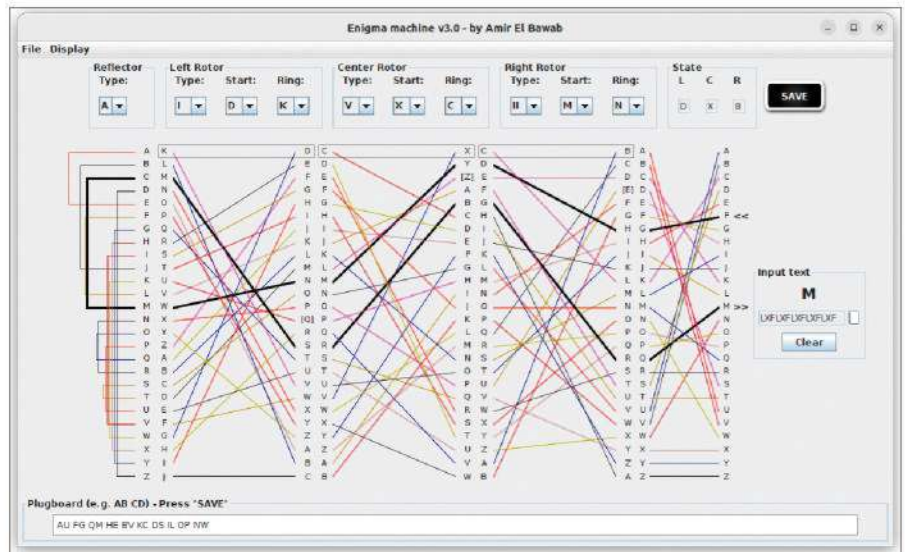
Virtual Enigma differs from many emulators due to its photo-realistic representation, even down to the radio transceiver and Morse key.

the bottom shows the patch leads connected to the plug board. Changing these from their defaults is intuitive so long as you have a basic understanding of the Enigma setup. Also at the top of the screen is something referred to as State, which shows the current positions of the left, centre and right rotors. This is provided because, unlike some emulators, the on-screen representation of the rotors doesn't change as they rotate. Connections in the plug board, rotors and reflector are shown in various colours, to make it easier to trace them by eye, and the current signal path is shown in bold black. However, there is no way of differentiating between signals en route to the reflector and signals that have been reflected. Having said that, the direction is fairly obvious, as the most recently typed letter is shown to the right of the wiring diagram.

Next up, if you want a bit of eye candy, how about taking a look at the photo-realistic Enigma emulator produced by Martin Gillow and called *Virtual Enigma*? It runs in a browser and is available at <https://enigma.virtualcolossus.co.uk/>. We're showing a screenshot (opposite page, bottom), so you know what to expect, but here's a brief overview. The opening view shows a typical Enigma operating setup, including the Enigma machine, the box containing alternative rotors, a sheet showing the required daily setup, and even a radio transceiver complete with Morse key. And because the Enigma operator would normally work alongside someone who would transcribe the encrypted or decrypted text on paper, there's a clipboard window that keeps a record of these text streams.

Setting up the Enigma is more realistic than using the menus of our previous emulator but, as a result, it's more complicated. Fortunately, there's extensive online help that you can view while using the emulator. In a nutshell, though, mouse clicks open the Enigma's case, open the cover over the rotors, remove the rotors and replace them with rotors from a box, configure the plug board with patch leads, and so much more.

If you tire of manipulating an online 3D Enigma emulator – however impressive it might be – you might wonder how much it would cost to get your hands on a real Enigma. We suspect, though, that you're not going to go beyond the wondering stage, given that they've been sold for tens if not hundreds of thousands of dollars, even approaching half a million in one case. However, if you're prepared to spend \$900, coming down to \$500 or \$600, for a kit of parts, you can build something that we're inclined to describe as looking Enigma-esque. These hardware emulators have a physical top panel with a keyboard and lamp board, and depending on which version you buy, there might also be a plug board. Hidden inside is an Arduino single-board computer. The main difference between this and a real Enigma is that there are no rotating wheels. Instead, there are four seven-segment LEDs, allowing it to emulate both the three-rotor and the four-rotor



Enigma – which indicate the position of each rotor. To learn more, head to www.stgeotronics.com.

Reliving Enigma needn't be an expensive undertaking, though. Admittedly, you're not going to get hands-on experience of using one but you can see them in several museums in the UK, including the National Museum of Computing at Bletchley Park, the Science Museum and the Imperial War Museum, both in London. Others are on display at museums in several other countries. And don't forget, should you visit Bletchley Park, you can also see working replicas of the bombe, which was used to defeat Enigma, and Colossus, which was instrumental in cracking messages encrypted by the Lorenz machine. **LXF**

This open source emulator by Amir El Bawab offers a clear view of the signal path through an Enigma machine.

» BEYOND ENIGMA

Despite it being the machine everyone has heard of, Enigma wasn't the pinnacle of cryptographic technology in WW2. Nor was it the most sophisticated cryptographic system that was cracked at Bletchley Park. That accolade goes to the German Lorenz machine.

Unlike Enigma, Lorenz was used for encrypting and decrypting teleprinter messages. Since these machines used the Baudot code – a binary code that we can think of as a 5-bit predecessor of ASCII – it had far more in common with current cryptography than Enigma. But while it shared Enigma's principle of using rotors, it had no fewer than 12 of them, thereby vastly increasing the key length.

Due to its vastly greater sophistication, instead of the electromechanical bombe machine that had been used to reveal the secrets of Enigma-encrypted messages, an electronic solution was developed for Lorenz. Called Colossus and developed at Bletchley Park, it's considered to have been the world's first programmable electronic computer. These were pre-transistor days, though, so Colossus was built using valves, no fewer than 2,400 of them in its second iteration, and it consumed 8.5kW. These were also pre-memory days, so the input data was read over and over, having been supplied on a continuous loop of paper tape. It might have been the first programmable electronic computer but it wasn't universal like today's machines. With the knowledge gained from wartime work, computers as we know them appeared just five years later.

» **IMPROVE YOUR LINUX SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

How to handle Nvidia graphics card drivers

Despite the company being famously anti-open source, **Neil Mohr** explores how to easily upgrade to an Nvidia graphics card.



**OUR
EXPERT**

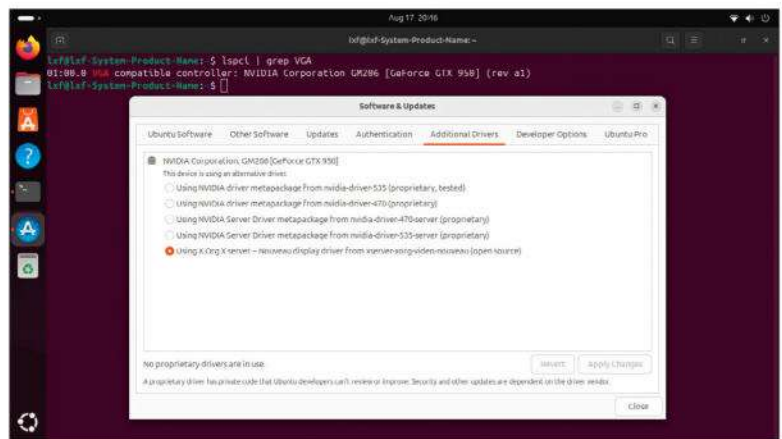
Neil Mohr remembers the delights of playing *Tomb Raider* on SLIed Voodoo 2 cards through OpenGL-based Glide back in the day.

It seems fair to say that Nvidia doesn't have the best reputation among Linux fans. If one picture could sum up a thousand words of the detest some developers have felt towards Nvidia, it would be the classic photo of Linus Torvalds middle-finger gesticulating to a crowd while uttering a favourite four-word profanity. Somewhat ironically, it's Nvidia's sheer success as a peddler of AI-powering GPUs that has forced it to soften its proprietary approach to its driver distribution – the issue being that nearly the entire AI world runs Linux servers and having to constantly post-install proprietary Nvidia drivers after kernel updates was costing time – and as we know, that's money. So now even Linus says they're not all bad...

But you don't have to be a server-farm-running sysadmin to fall foul of Nvidia driver issues – plenty of home users have suffered problems post-system upgrade for similar reasons, although getting drivers in the first place can be befuddling. This is partly due to multiple sources being available, different install routes being offered by different distros, not knowing which driver version you should be running and, of course, Wayland sticking its oar in.

So, this month we are having a quick look at what to check before installing an Nvidia graphics card, the easy distros' installation routes, the Nvidia installer, and how to remove Nvidia drivers if something goes wrong. Then we look at six of the best cards over the page.

To start, the reason for Nvidia hate is that it's only offered a proprietary driver for years. This means it can't be included in the GPL-licensed Linux kernel. Nvidia's workaround is to supply a GPL-compliant software 'shim' that is GPL licensed. This enables the proprietary driver to be installed afterwards and work seamlessly with the kernel. Problems arise, however, when kernel updates require the Nvidia kernel modules to be rebuilt against the new kernel headers. If there are any problems, the driver won't load on reboot and



The user-friendly GUI interface that should offer all the driver options you need.

that can lead to there being no display. In the past, the workaround was to purge Nvidia drivers before kernel updates; thankfully, these days updates tend to go more smoothly.

Before starting, we should mention Nouveau. This is the project that develops an open source driver for the Linux kernel that can be included by default. "So why don't we all use that?" you cry. Because, despite valiant reverse-engineering efforts, it lags behind the official drivers when it comes to supporting new hardware, and it can't match them for 3D performance. So, it's fine for desktop use and, as of November 2023, it added support for AD10x into Linux kernel 6.7, which is the RTX 40x0 range of consumer cards.

Check it...

Even if you plan to use your distro's driver support or package manager to install the Nvidia driver, it's well worth checking what the current Nvidia-supported driver version is, even if you don't download it. Head to www.nvidia.com/download/index.aspx – you need to select the Type (usually GeForce), the series (as in 40 for RTX 40x0) and then Linux.

The Download Type offers Production Branch and New Feature Branch. We urge you to select Production, because this is the stable tested build; New Feature Branch is really the beta release. Click Search and it provides details of the most suitable

GPU

Upgrade it: Nvidia GeForce graphic cards

Picking the best graphics card can be confusing, so **Jarred Walton** cuts through the jargon and helps us choose.

The graphics card is the beating heart of any gaming PC, and everything else comes second. Without a powerful GPU pushing pixels, even the best CPUs for gaming won't manage much. No one graphics card is right for everyone, so we've provided options for every budget and mindset. Whether you're after the fastest card, the best value, or the best at a given price, we've got you covered.

Many people have a price bracket in mind when choosing a graphics card and simply want the best price for performance. So, we've chosen the six best value per frame Nvidia cards from the 10 available, looking at 1080p, 4K and ray-tracing results. We're focusing on Nvidia here, we'll look at AMD soon.

Where often GPU benchmarks rank cards based purely on performance, our list of the best graphics cards here looks at the whole package. Current GPU pricing, performance, features, efficiency and availability are all important, though the weighting is more subjective. Factoring in all of those aspects, these are the best graphics cards currently available.

On the horizon

We've entered the calm before the storm, as far as desktop GPUs are concerned. Nvidia Blackwell RTX

50-series GPUs should arrive later in 2024, with AMD RDNA 4 and Intel Battlemage also in the works, but none of those are out yet, so we're in a holding pattern until they arrive. If you're after a high-end card, you might be best waiting for these new arrivals.

We haven't had a 'new' GPU launch since February's AMD RX 7900 GRE, joining the four GPUs that came out in January: RTX 4080 Super, RTX 4070 Ti Super, RTX 4070 Super, and AMD RX 7600 XT.

Barring a surprise announcement, this should be the end of new models until the future Nvidia Blackwell and RTX 50-series, AMD RDNA 4 and Intel Battlemage GPUs arrive, although that remains to be seen. The closest thing to new GPUs is Nvidia's SFF-ready GPU specifications, but those likely sacrifice some performance to fit into a smaller volume.

Prices are mostly sitting at around their original RRP at the high end and below for the least expensive offering on each GPU. That includes the RTX 4080 Super, though the RTX 4090 costs £1,800 right now – so you may as well wait for the RTX 5090 at this point. That advice applies to most of the high-end cards: If you haven't already upgraded, waiting until November to see what becomes available makes a lot of sense especially at the prices still being commanded.

We've provided core benchmarks for Nvidia's current cards, and selected the six best to look at in more detail.

GPU	1080p FPS	£ per FPS	1080p ray tracing	£ per FPS	4K FPS	£ per FPS	Price	Average power
GeForce RTX 4090	190.1	£9.47	121.4	£14.83	112.8	£15.96	£1,800	310W
GeForce RTX 4080 Super	163.5	£6.06	97	£10.21	87.7	£11.29	£990	240W
GeForce RTX 4080	160	£6.50	94.8	£10.97	84.4	£12.32	£1,040	241W
GeForce RTX 4070 Ti Super	142.4	£5.55	83	£9.52	73.9	£10.69	£790	244W
GeForce RTX 4070 Ti	134.1	£5.44	80.4	£9.08	67.4	£10.83	£730	227W
GeForce RTX 4070 Super	124.2	£4.67	70.9	£8.18	62.6	£9.27	£580	194W
GeForce RTX 4070	106.7	£4.69	63	£7.94	53.9	£9.28	£500	179W
GeForce RTX 4060 Ti 16GB	85.4	£5.04	45.5	£9.45	41	£10.49	£430	148W
GeForce RTX 4060 Ti	83.6	£4.25	46.3	£7.67	40.9	£8.68	£355	139W
GeForce RTX 4060	66.7	£4.20	36.1	£7.76	33.5	£8.36	£280	124W

GeForce RTX 4070 Super

SPECS

GPU: AD104
Compute: 56
Shaders: 7,168
Tensor: 224
RayT: 56
Clock:
 2,475MHz
RAM: 12GB
 GDDR6, 21Gb/s
L2 cache:
 48MB
TFLOPS: 35.5
 FP32, 284
 FP16, 568 FP8
TGP: 220W
Launch: Jan 24

Nvidia refreshed its 40-series line-up at the start of 2024 with the new Super models. Of the three, the RTX 4070 Super will likely be of interest to the most people. It inherits the same £589 RRP as the non-Super 4070 (which has dropped to around £500 to keep it relevant), with all the latest features of the Nvidia Ada Lovelace architecture. It's slightly better than a linear boost in performance relative to price, which is as good as you can hope for these days.

There appear to be plenty of RTX 4070 Super base-RRP models available at retail. We like the stealthy black aesthetic of the Founders Edition, and it runs reasonably cool and quiet, but third-party cards with superior cooling are also available – sometimes at lower prices than the reference card.

The 4070 Super bumps core counts by over 20% compared to the vanilla 4070, and in our testing we've found that the general lack of changes to the memory subsystem doesn't impact performance as much as you might expect. It's still 16% faster overall (at 1440p), even with the same VRAM capacity and bandwidth – though helped by the 33% increase in L2 cache size.



If people made sensible choices, this is where they'd end up.

Compared to the previous generation Ampere GPUs, even with less raw bandwidth, the 4070 Super generally matches or beats the RTX 3080 Ti, and delivers clearly superior performance compared to the RTX 3080. What's truly impressive is that it can do all that while cutting power use by over 100W.

VERDICT

DEVELOPER: Nvidia **PRICE:** £589

» **Rating 9/10**

GeForce RTX 4070

SPECS

GPU: AD104
Compute: 46
Shaders: 5,888
Tensor: 184
RayT: 46
Clock:
 2,475MHz
RAM: 12GB
 GDDR6, 21Gb/s
L2 cache:
 36MB
TFLOPS: 29.1
 FP32, 233
 FP16, 466 FP8
TGP: 200W
Launch: Apr 23

Nvidia's RTX 4070 didn't blow us away with performance or value, but it's generally equal to the previous-generation RTX 3080, comes with the latest Ada Lovelace architecture and features, and now costs about £180 less. With the launch of the RTX 4070 Super (see above), it received a further price cut and the lowest-cost cards now start at around £500.

Nvidia rarely goes after the true value market segment; however, with the price adjustments brought about with the arrival of the recent 40-series Super cards, things are at least reasonable. The RTX 4070 can still deliver on the promise of ray tracing and DLSS upscaling, it only uses 200W of power (often less), and in raw performance, it outpaces AMD's RX 7800 XT – slightly slower in rasterisation, faster in ray tracing, plus it has DLSS support.

Nvidia is always keen to point out how much faster the RTX 40-series is once you enable DLSS 3 Frame Generation, but this isn't available to Linux, and Nvidia isn't saying whether it ever will be. Even so, if it were, we'd say DLSS 3 would improve the experience over the baseline by perhaps 10-20%.



If you can't afford the Super, this is second best.

The choice between the RTX 4070 and the above RTX 4070 Super comes down to whether you're willing to pay more for proportionately higher performance – the two offer nearly the same value proposition otherwise if you look at the charts. The only standout point is the RTX 4070's slightly lower power draw, but even so, £500 is a lot to sink into a single card.

VERDICT

DEVELOPER: Nvidia **PRICE:** £500

» **Rating 8/10**

GeForce RTX 4090

SPECS

GPU: AD102
Compute: 128
Shaders: 16,384
Tensor: 512
RayT: 128
Clock: 2,520MHz
RAM: 24GB
 GDDR6, 21Gb/s
L2 cache: 72MB
TFLOPS: 82.6
 FP32, 661 FP16,
 1,321 FP8
TGP: 450W
Launch: Oct 22

For some, the best graphics card is the fastest card, pricing be damned. Nvidia's GeForce RTX 4090 caters to precisely this category of user. It was also the debut of Nvidia's Ada Lovelace architecture and represents the most potent card Nvidia has to offer, likely until later this year when the next-generation Blackwell GPUs are set to arrive.

Note also that pricing of the RTX 4090 has become extreme, with many cards now selling above £2,000. That's due to a combination of factors, including China RTX 4090 export restrictions and the rise of AI.

The RTX 4090 creates a larger gap between itself and the next closest Nvidia GPU. Across our suite of benchmarks, it's 35% faster overall than the RTX 4080 at 4K, and 32% faster than the RTX 4080 Super. It's also 47% faster than AMD's top-performing RX 7900 XTX – but it also costs nearly twice as much.

Let's be clear: you really need a high refresh rate 4K monitor to get the most out of the RTX 4090. At 1440p, its advantage over a 4080 Super shrinks to 22%, and only 16% at 1080p – and that includes demanding DXR games. You'll also want the fastest CPU possible.

It's not just gaming performance; in professional content creation workloads like *Blender*, *Octane* and *V-Ray*, the RTX 4090 is up to 42% faster than the RTX 4080 Super. And with *Blender*, it's over three times



You'll be the fastest kid on the block and also the poorest.

faster than the RX 7900 XTX. Don't even get us started on AI tasks. In *Stable Diffusion* testing, the RTX 4090 is also around triple the performance of the 7900 XTX for 512x512 and 768x768 images.

There are other AI workloads that only run on Nvidia GPUs. The only potential snag is that it uses drivers to lock improved performance in some apps to its true professional cards – the RTX 6000 Ada Generation.

AMD's RDNA 3 response to Ada Lovelace might be a better value, but for raw performance, the RTX 4090 reigns as the current champion. Just keep in mind that you may also need a CPU and power supply upgrade.

VERDICT

DEVELOPER: Nvidia **PRICE:** £1,800

» **Rating 8/10**

GeForce RTX 4060

SPECS

GPU: AD107
Compute: 24
Shaders: 3,072
Tensor: 96
RayT: 24
Clock: 2,460MHz
RAM: 8GB
 GDDR6, 17Gb/s
L2 cache: 24MB
TFLOPS: 15.1
 FP32, 121 FP16,
 242 FP8
TGP: 115W
Launch: July 23

With the launch of the RTX 4060, Nvidia appears to have gone as low as it plans for this generation of desktop graphics cards based on the Ada Lovelace architecture.

There are drawbacks with this level of GPU. Nvidia opted to cut the memory interface to just 128 bits, which limits the memory capacity options. Nvidia could do a 16GB card if it wanted, but 8GB is the standard and we don't expect anything else – outside of its professional GPUs, only the 4060 Ti 16GB has the doubled VRAM option, and we weren't particularly impressed by that card. The 4060-class cards also have an x8 PCIe interface, which shouldn't matter much, although it might reduce performance if you're on an older platform that only supports PCIe 3.0.

Nvidia also cut down the number of GPU cores on the RTX 4060 compared to its RTX 3060 ancestor. The 3060 had 28 SMs (Streaming Multiprocessors, with 128 CUDA cores each); the 4060 has 24. Higher clocks help, but the RTX 4070 and above all have at least the same number of SMs as their predecessors.

The good news is that, as promised, performance is at least faster than the previous-generation RTX 3060, by about 20% at 1080p and 1440p. There are edge cases in some games (4K at max settings) where the 12GB on the 3060 can pull ahead, but performance is already well below the acceptable level at that point.

Your wallet is going to love you, but not all of your games will.

There are other benefits with the 4060. You get all the latest Ada features, plus the power draw is just 115W for the reference model, and typically won't exceed 125W on overclocked cards.

AMD's closest alternative is the previous-generation RX 6700 XT: higher rasterisation performance from AMD, worse ray tracing (though the point remains on just how useful this is), and higher power needs.

An alternative view is that this is an upgraded RTX 3050, with the same 115W TGP and 60% better performance. Too bad it costs slightly more – though the 3050 was mostly priced at £250 and above until later in life, so this does remain a solid budget pick.



VERDICT

DEVELOPER: Nvidia **PRICE:** £280

» **Rating 7/10**

GeForce RTX 4080 Super

SPECS

GPU: AD103
Compute: 80
Shaders: 10,240
Tensor: 320
RayT: 80
Clock: 2,550MHz
RAM: 16GB
GDDR6, 23Gb/s
L2 cache: 64MB
TFLOPS: 52.2
FP32, 418 FP16, 836 FP8
TGP: 320W
Launch: Jan 24

The RTX 4080 Super at least partially addresses one of the biggest problems with the original RTX 4080: the price. The 4080 cost £1,450 at launch, a 120% increase in generational pricing compared to the RTX 3080. The 4080 Super refresh costs £990, representing a drop in generational pricing compared to the RTX 3080 Ti. The 3080 Ti felt overpriced, but it nearly matched the 3090 in performance, just with half the VRAM. The 4080 Super still trails the 4090 by 24% at 4K.

With the current price premiums on the RTX 4090, likely caused by AI companies buying up GPUs, we've also seen increased pricing on the RTX 4080 Super — many cards cost above RRP, though at least a couple of RRP-priced models are available. If you're interested, check out the PNY RTX 4080 Super and Zotac 4080 Super at £985. That's £815 less than the general 4090 pricing, if you're keeping track.

On paper, the 4080 Super is better than the now discontinued vanilla 4080, even if the upgrades are minor. The 4080 Super increases core counts by 5%, GPU clocks by 2%, and VRAM clocks by 3% — a net 3% increase in performance. Faster and less expensive would be moving in the right direction.

Nvidia also goes directly after AMD's top GPU with the 4080 Super, and with the price reduction, it becomes a far more compelling choice. Overall, the



This has 80% of the speed of a 4090 but for half the price.

4080 Super leads the 7900 XTX by 14-16% at 4K and 1440p, and even if we only look at rasterisation, it's tied or slightly faster than AMD's best. Paying £90 extra for Nvidia's faster and more efficient card, plus getting the other Nvidia extras, should be an easy decision.

It's not all smooth sailing for the 4080 Super, given the current availability and prices. Our advice is to only pick up a card if it costs closer to £1,000. Otherwise, there are better options. You could also wait for the RTX 5080 replacement that will likely land in October/November, but predicting pricing and performance of such GPUs is beyond the capabilities of our crystal ball.

VERDICT

DEVELOPER: Nvidia **PRICE:** £990

» **Rating 7/10**

GeForce RTX 4060 Ti

SPECS

GPU: AD106
Compute: 34
Shaders: 4,352
Tensor: 136
RayT: 34
Clock: 2,535MHz
RAM: 8GB
GDDR6, 18Gb/s
L2 cache: 32MB
TFLOPS: 22.1
FP32, 177 FP16, 353 FP8
TGP: 160W
Launch: May 23

What is this, 2016? Nvidia's RTX 4060 Ti was a £399 graphics card, with only 8GB of memory and a 128-bit interface — or the 4060 Ti 16GB doubles the VRAM on the same 128-bit interface, for about £50 extra. We thought we had left 8GB cards in the past after the RTX 3060 gave us 12GB, but Nvidia seems intent on cost-cutting and market segmentation these days. The RTX 4060 Ti does technically beat the previous generation RTX 3060 Ti, by 10-15% overall.

The larger L2 cache does mostly overcome the limited bandwidth from the 128-bit interface, but cache hit rates go down as resolution increases, meaning 1440p and 4K can be problematic. At least the price is the same as the outgoing RTX 3060 Ti; street prices start at £355 now, and you do get some new features. But 8GB feels very stingy, and the 16GB model mostly helps at 4K ultra, where performance is still poor.

As with the RTX 4060 (left), Nvidia reduced the number of GPU cores on the RTX 4060 Ti compared to its RTX 3060 Ti predecessor. The 3060 Ti had 38 SMs while the 4060 Ti only has 34. Again, ~45% higher GPU clocks make up for the 11% core count deficit, but the reduction in memory interface width and GPU cores does feel incredibly stingy.

There's not nearly as much competition in the mainstream price segment as we would expect. AMD's RX 7700 XT is up to 10% faster than the 4060 Ti but



Triple 1080p frame rates and playable 4K make this a keen choice.

costs about 15% more. It's 8% faster than the 16GB variant for basically the same price, but it's the usual mix of winning in rasterisation and losing in ray tracing.

The 4060 Ti manages 1440p ultra at 60fps in rasterisation games, but for ray tracing you'll want to stick with 1080p — or use DLSS, but as mentioned, the Frame Generation that interpolates frames isn't available on Linux and may never be.

The 4060 Ti — both 8GB and 16GB variants — are worth considering but there are better picks available from AMD for pure rasterisation performance. **LXF**

VERDICT

DEVELOPER: Nvidia **PRICE:** £355

» **Rating 7/10**

Fixing system services on Linux servers

Stuart Burns laments the fact that fixing broken services is much easier on Linux than in the UK at large.

For many people who don't usually get involved in the intricacies of managing Linux, troubleshooting services isn't always straightforward. Even for professionals, it can be a little challenging at times. Here we hope to help impart some knowledge of how to approach the management of services in Linux.

When a service fails, it disrupts operations and affects end users, including the potential costs and downtime in a business scenario that might be involved. Understanding how to manage and troubleshoot services is essential for maintaining system stability, security and performance.

Understanding Linux services

Before discussing troubleshooting, it's important to understand what a Linux service is. A service is typically a long-running process that performs specific tasks in the background. These services can include web servers (such as Apache or Nginx), database servers (such as for MySQL or PostgreSQL), or system daemons, such as *cron* or even *sshd*.

In Linux, services are managed by an init system. *Systemd* is by a wide margin the most popular, with most major distros using it, so that is what we are covering here.

A service consists of three major components:

- **Init systems:** Init systems are used to manage the startup, shutdown and management of services. You can think of these as the commands used to manage the daemons.
- **Daemon:** The managed background process that runs without direct user interaction – for example, Apache. Daemons can usually be run directly from the command line, but it's not advised. Let's do it properly!
- **Unit files:** Configuration files used by *systemd* to define how services should be managed (including setting whom the user runs as, the location of key files and so on).

To see an example of what services are managed by *systemd*, you can use the command:

```
$ sudo systemctl list-unit-files
```

Notice how some services are disabled. This is quite typical and nothing to worry about.

Basic services management

When services are behaving, they are quite easy to manage. To give a few pertinent examples:

```
$ systemctl start <service> # Starts service in question
```

```
$ systemctl stop <service> # Stops service in question
```

```
$ systemctl enable <service> # Configures service to run at boot time.
```

```
$ systemctl is-enabled <service> # Is service set up to start at boot?
```

```
$ systemctl is-active <service> # Is service running?
```

When a service fails, it can be due to several reasons. Common issues include:

- **Service misconfiguration:** Incorrect settings in configuration files.
- **Dependency failures:** Services often rely on other services or system resources, and if these are unavailable, the service might fail.
- **Permission issue:** Incorrect file or process permissions can prevent a service from running.

The troubleshooting process

When troubleshooting services, a systematic approach is essential to identify and resolve the root cause.

- **Verify the service status:** The `systemctl status` command provides detailed information. For example, if Apache is installed, its status can be checked using the following:

```
$ systemctl status httpd
```

This command tells you whether the service is active, inactive or failed, along with recent logs.

A useful *systemctl*-related command is `journalctl -xe`. Yes, it's another 'ctl' command to master, but it is useful to get more details on why a specific daemon failed. It is our initial go-to after a service failure as it usually points to the next place to look.

By default, services are not configured to start on boot automatically (that is achieved via `systemctl enable <service>` and `systemctl start <service>`).

Other things to check out include `sudo dmesg`, which gives a lot of information about everything occurring on the host, not just the services being investigated. Other log locations can be found under `/var/log` (it all depends on where the system is configured to log to).

Using the status command shows that our Apache is currently dead.

```
[sysadmin@headnode ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
```

● **Check service log:** Most services log their activity to specific log files. For example, Apache logs are typically found in `/var/log/apache2/`. Use `tail` or `less` to inspect these logs:

```
$ tail -f /var/log/apache2/error.log
```

● **Inspect configuration files:** If the service fails to start or behaves incorrectly, there might be an issue with its configuration files. For example, for Apache, you would check the `/etc/apache2/` directory. Use a text editor like `vim` or `nano` to open and inspect these files. Pay attention to syntax errors, incorrect paths or invalid directives.

● **Test dependencies:** Ensure that any dependencies the service relies on are functioning correctly. For example, if a web service depends on a database, ensure the database service is running and accessible. Use tools like `ping`, `telnet`, or `nc` to test connectivity. That said, most standalone services should at least log this as an issue.

● **Review system logs:** System-wide logs can provide insights into issues that may not be directly related to the service but are affecting it. Check `/var/log/syslog` or `/var/log/messages`, depending on your distribution.

● **Verify the service has the correct file permissions and ownership:** Services may fail to start or function correctly without appropriate permission to access necessary files. Use `ls -l` to check permissions:

```
$ ls -l /path/to/file
```

● **Restart the service:** Sometimes, restarting the service can resolve transient issues. Use `systemctl restart` on `systemd`-based systems:

```
$ systemctl restart apache2
```

● **Reboot:** As a last resort, especially if multiple services fail, rebooting can resolve issues caused by memory leaks, locked resources or stuck processes.

Troubleshooting services in Linux requires a deep understanding of how services work, as well as a systematic approach to diagnosing and resolving issues. By following a structured process and utilising the right tools, system administrators can effectively address service disruptions and maintain the stability and performance of their systems. With practice, these troubleshooting skills become second nature. **LXF**

UNIT	FILE	STATE
proc-sys-fs-binfmt-misc-automount		static
-.mount		generated
boot.mount		generated
dev-hugepages.mount		static
dev-mqueue.mount		static
proc-sys-fs-binfmt-misc.mount		static
sys-fs-fuse-connections.mount		static
sys-kernel-config.mount		static
sys-kernel-debug.mount		static
tmp.mount		disabled
systemd-ask-password-console.path		static
systemd-ask-password-plymouth.path		static
systemd-ask-password-wall.path		static
session-1.scope		transient
session-3.scope		transient
auditd.service		enabled
autovt@.service		enabled
blk-availability.service		disabled
console-getty.service		disabled
container-getty@.service		static
cpupower.service		disabled
crond.service		enabled
dbus-org.fedoraproject.FirewallD1.service		enabled
dbus-org.freedesktop.hostname1.service		static
dbus-org.freedesktop.locale1.service		static
dbus-org.freedesktop.login1.service		static
dbus-org.freedesktop.nm-dispatcher.service		enabled
dbus-org.freedesktop.portable1.service		static
dbus-org.freedesktop.timedate1.service		static
dbus.service		static
debug-shell.service		disabled
dm-event.service		static
dnf-makecache.service		static
dnf-system-upgrade-cleanup.service		static
dnf-system-upgrade.service		disabled
dracut-cmdline.service		static
dracut-initqueue.service		static
dracut-mount.service		static
dracut-pre-mount.service		static
dracut-pre-pivot.service		static
dracut-pre-trigger.service		static
dracut-pre-udev.service		static
dracut-shutdown-onfailure.service		static

lines 1-44

This shows which services are running on the host. It is possible to use `grep` pipe to limit output.



Stuart Burns is a Linux administrator for a Fortune 500 company specialising in Linux.

» RUNNING THE RISC

It can't be denied that I am a bit of a Raspberry Pi supporter (and shareholder!). Much of the press and the wider Linux community failed to notice something different about the latest Pico. Rather than featuring a pure Broadcom ARM-based CPU, it includes a pair of RISC-V CPU cores embedded in the chip. OK, it's not powerful but it's cheap and plentiful. This is a brave and novel approach to testing the waters for alternative CPUs.

Other companies sell RISC-V based systems but not in an inexpensive package backed by a well-known company with good engineering resources.

Don't forget, all the hobbyists love it, but by far the larger Pi market is companies that integrate them into their products as a cheap and easily replaceable SBC.

Pi aside, the other interesting news is that Canonical has decided to be more proactive about the kernel it includes in its LTS releases. Canonical has declared that it will use RC (Release Candidate) kernel releases if they are available when an LTS edition ships, rather than staying with the old kernel that is well tested.

The logic behind this is that because of the long-term nature of LTS releases that ship an older kernel, that kernel persists for a long time, leaving out the useful new features and base hardware support included in the latest kernel.

Don't worry, though – as that kernel gets updated to non-RC status, those new releases will percolate into the update channels. It's not like you'd be stuck on pre-release kernels for the life of the distribution.

» PRO TIP CORNER!

Any admin knows how annoying it can be to always prefix commands with `sudo`, especially when you do it hundreds of times a week. One quick tip is to switch to root by using the command `sudo -i`. This elevates the user to the root account so commands can be run as root. No more having to prefix with `sudo` all the time.

What makes it particularly useful is that your sysadmin may

have prevented `sudo su` as a way to switch to root (assuming appropriate rights have been granted) `sudo -i` addresses this issue and still gets properly logged in the log files.

Somewhat ironically, there is now talk of replacing `sudo` with an updated setup that uses `systemctl`. Only time will tell how that works on one of the most frequently used commands on a modern Linux system.

IN DEPTH Inside the AI PC

Inside the AI PC

Is the AI PC anything more than marketing speak? **Barry Collins** and **Tim Danton** find out...



Remember when 3D screens were the next big thing? What about virtual reality? Or 5G? The truth is that the technology industry loves buzzwords almost as much as it loves the tech itself, and its latest crush is AI PCs.

The trouble is, if you ask five different vendors what an AI PC is, you would likely get different answers. AMD and Intel will explain that it's all about their new silicon, which now includes neural processing units to go along with CPUs. Nvidia will point to its hugely powerful RTX GPUs. Microsoft will tout *Copilot*.

Truth is, the definition is hazy. By many definitions, you already have an AI PC to hand via your mobile phone. It's personal, it computes, and if it's a recent Apple, Google or Samsung phone, it includes plenty of AI-enhanced features. Magic Eraser is one example.

So, let's lay down a simple working definition. An AI PC – for the duration of the next pages, at least – is a laptop or desktop PC that includes a processor with an NPU. That means an Intel Core Ultra, any AMD Ryzen processor with Ryzen AI built in, and Apple MacBooks and Mac desktops with an M1, M2 or M3 series chip – but what about poor penniless Nvidia?

With that settled, let's get to work. What's the point of an AI PC? What can it actually do? What options are available? And, most importantly, should you buy one?

AI PCS NOW!

If you're in the market for a new PC and are weighing up whether to invest in an AI PC, you might wonder what advantages it will bring over an 'ordinary' PC.

We'll come to the benefits afforded by the piece of hardware that pretty much defines an AI PC – the neural processing unit (NPU) – shortly, but there's another distinguishing piece of hardware found on many AI PCs: the Microsoft Copilot key.

The Copilot key was billed as the first major change to PC keyboard layouts in almost 30 years, replacing the menu key to the right of the spacebar. It's basically a shortcut for opening the *Copilot* assistant in Windows 11 and is largely redundant anywhere else.

Dell's clearly excited about having a new key. "We really think of *Copilot* as the new search," said Kevin Terwilliger, VP and general manager of Dell's Latitude

and Docking business. "People are going to [interact] with *Copilot* throughout their day, engaging with it, asking questions, working with it to create content."

Let's not fool ourselves, Microsoft is going to be driving *Copilot* integration and marketing hard over the next year or two – it's plunged \$13 billion into OpenAI to date. At the time of writing, *Copilot*'s ability to control the PC is pretty limited. However, Windows Insider releases show Microsoft continuing to build out *Copilot*'s system-level capabilities.

The question for Linux users is whether open source developers are planning to integrate similar AI features into the Linux desktop? Past fads such as accelerated search, app stores, docks, animated wallpaper and more have made it into open source desktops, so we imagine it's just a matter of time.

WHAT IS AN NPU?

The NPU is the core component of an AI PC; it's what really sets it apart from a PC that only has CPU and GPU units to call upon. Over time, it's likely all PCs will ship with an NPU, but for now, they're the key differentiating factor.

But if you're an early adopter, one who rushed out to buy a laptop or PC with an NPU-blessed Intel Core Ultra or AMD Ryzen chip inside, what benefits will you feel right now, today? Even the PC manufacturers

The Copilot key can be found on most AI PC laptops, including this Asus Vivobook S15 Copilot.



» AN AI TO CALL YOUR OWN...

It's a misconception that you need the huge processing power of cloud servers to run LLMs. It's possible to run LLMs locally on even relatively modestly powered PCs, even without an NPU or dedicated graphics card.

If you've ever tried to download and run an LLM locally, you may have been put off by the need for prerequisites, environments and web UIs to make it all work. However, there is an easier way.

The free *LM Studio* (<http://lmstudio.ai>) for Linux, Mac or Windows lets you download and run LLMs without any of the faff. You simply choose which LLM

you want to run, by either picking from a selection on the home screen or typing its name into the search bar, and it downloads and runs it.

At the time of writing, available LLMs included Google's recently released *Gemma*, *Qwen* from the Alibaba group, and *Code Llama* from Meta. *LM Studio* gives handy descriptions of the LLMs featured on the homepage to help you choose, as well as details of how much memory they require and the size of the download. Most LLMs are 2-10GB.

Once you've made your choice, click on the Chat icon on the left, select the

model you want to interact with and start chatting away. (If you can't see the User chat field, close the download pane in the bottom half of the screen to reveal it.)

On the right-hand side, you should be able to access the LLM's settings. Here you can normally enter a system or pre-prompt that tells the LLM how you want it to behave. For example, "Give answers as if explaining to a five-year-old" or "Assume Linux technical knowledge." You can also choose the level of GPU acceleration, prompt overflow settings and more.

quietly concede the advantages are slim pickings at the moment.

One that crops up often in PC manufacturer demos is the NPU's ability to blur the background on video calls. Most users won't care if it's the CPU, GPU, NPU or another TLA that performs something so mundane, but it can make a difference to power consumption.

Dell ran a demo showing background blur being applied on a Zoom call on an ordinary PC, where CPU utilisation shot up to more than 8% when the background blur was applied. By applying the same effect locally on the NPU, using *Windows Studio Effects*, the CPU was barely troubled at 1% utilisation. "This translates into a 38% power improvement when you're doing these Zoom calls," claimed Terwilliger.

Dell also points to security benefits, claiming the NPU is better placed to deal with threat detection. The company ran a demo of an unnamed security package, but instead of running threat detection in the cloud, it used a local engine on the NPU. Dell claims detection on the AI PC kicked in within 20 milliseconds, a 70% improvement compared to the cloud. "Our goal is to not talk about the hype of AI, but show examples just like this to our customers so that they can understand the value of the NPU," said Terwilliger.

You're not racing off to the Dell website to order an AI PC yet? Maybe a more creative application will convince you, and here the benefits are arguably more substantial. Instead of relying on expensive cloud services such as OpenAI's *DALL-E* or *Midjourney* to generate images, you could install open-source models such as *Stable Diffusion* and run them locally instead.

You can, of course, do that on a PC without an NPU. But, claims Dell's senior director and general manager for Precision workstations, Charlie Walker, it's much faster with the NPU involved. "If I tried to run *Stable Diffusion* on my CPU, previously it would have taken two to three minutes [to generate an image]," he said. "Now with the NPU, by offloading that from the CPU into that more efficient architecture ... you can do that now in 30 to 50 seconds. So again, significant savings."

AN AI-POWERED FUTURE

"What you're doing on an AI PC in 2024 is going to be radically different to 2026," said Intel's executive VP for client computing, Michelle Johnston Holthaus at CES 2024. The company's rival agrees. "It will probably be the end of 2025 before you really start to see what an AI PC is capable of," said AMD's senior processor technical marketing manager, Donny Woligroski.

"The GPU is currently the fastest AI processing unit," Woligroski added. "You're going to see that performance cross over, with the NPU doing the same levels of [AI] performance as the GPU."

Lenovo agrees the best is yet to come from AI PCs, pointing out that the software developers have only just started working with production hardware. "It's almost a chicken and the egg," said Tom Butler, the company's executive director of commercial portfolio and product management. "I've got to have hardware out there that can catch the software benefits. The front end of this wave is hardware-driven, because there was nothing for the software to write to, because a lot of this was in the labs of the silicon providers and

OEMs like Lenovo. We couldn't go out broadly and have these discussions with software companies."

Now, according to Butler, "We're actively in conversations with a broad breadth of software companies. Basically everybody's looking at this new space to see 'what can I do now?'"

The pace of development has been nothing short of breakneck. Subscribe to the Ben's Bites newsletter (<http://bensbites.beehiiv.com>), for example, and you'll find multiple new AI product launches announced every day. Butler predicts you won't have to wait too long to see exciting new product launches that tap the capabilities of AI PCs, either: "I think if you just project forward near term – like in months, quarters, not years – you're going to see a lot more capability coming."

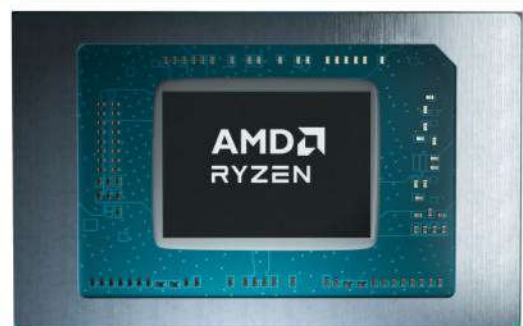
Dell's Terwilliger agrees that we're only just getting started with AI PC apps. "We're just at the tip of the spear," he said. "Intel has talked about the 100+ ISVs [software developers] that are utilising the Core Ultra processor. [W]hat we also see, companies like Dell, large organisations, are developing new AI capabilities to roll out to end users. That's going to take advantage

» AI-POWERED CPUS

We've all become used to cores and gigahertz as the key stats for processors, but as we slide deeper into 2024, that is set to change. Now, the silicon makers are starting to talk about TOPS, which stands for trillions (or tera) operations per second. The sheer magnitude of the number here gives you some idea of AI PCs' potential power, because – in the right situation – they can achieve so much in a fraction of a second.

Here, we provide a look at what AMD, Apple, Intel and Qualcomm have to offer.

AMD RYZEN 7040 AND 8040 SERIES



7040: 33 TOPS, 8040: 39 TOPS

AMD beat Intel to the AI punch by almost a year, with its CEO Lisa Su announcing the Ryzen 7040 series back in January 2023 and the first laptops available by March. Then in December it announced the updated 8040 series, with the promise of a slight boost in performance. It's a little confusing, as not every 7040 and 8040 series chip includes Ryzen AI – AMD's name for its NPU – so you should check each processor's listing before buying.

of this NPU to be able to run efficiently and not have to move a bunch of data into the cloud.”

“If you think about AI now, most of it is cloud based,” said Butler. “That opens up latency concerns, security and privacy concerns. The ability to bring that down to the edge device not only overcomes some of those hurdles, but makes it more personal, you’re working on your work for you, not a public cloud-based platform.”

One advantage of local AI software is that it could access apps and data piles stored on your PC, rather than being restricted to files you upload to the cloud. Butler said you could think of this as an “orchestration layer”, capable of bringing together the capabilities of different apps, working with system-wide data. “It’s almost a prompt-led conversation with your system, not ‘let me open this app to accomplish a task’.”

He envisages a future where you’ll tell the AI you want to accomplish a specific task and it will tell you the most efficient path to accomplish that using the resources and apps available on your PC. “That’s not present today, but that’s effectively what you want to drive to from an AI perspective,” he said. **LXF**

Running an AMD Ryzen 7840HS processor makes the Framework an AI PC.



APPLE M3



18 TOPS (NPU only)

When it comes to NPUs being built into silicon, Apple has a three-year head start. When the M1 appeared in 2020, it already included one, and while Apple didn’t state its power, the internet’s best guess (from @t3mporarybl1p on X) is 11.3 TOPS. It’s surprising, then, that the M3 is stuck on 18 TOPS. However, Apple claims that the new MacBook Air is the “world’s best consumer laptop for AI”. While that may not be true of the hardware, Apple can rely on a strong software ecosystem, not least because developers have been exploiting the Neural Engine for so long.

AMD RYZEN 8600G AND 8700G



39 TOPS

AMD has also beaten Intel to the desktop punch, announcing the Ryzen 8600G (see **LXF314** for a review) and 8700G chips at CES 2024 in January. Just to muddy things a little, the 8500G and 8300G – announced at the same time – don’t include Ryzen AI – see **LXF313** news for more on Ryzen AI. At £649, the PC Specialist Fusion Elite P, a PC based on the 8600G, is an extremely affordable entry into the AI PC world.

INTEL CORE ULTRA



34 TOPS

Launched in December 2023, Intel’s Core Ultra chips were the company’s first to feature an NPU. Only 11 Core Ultra processors have been released so far, ranging from the Ultra 5 125U to the Ultra 9 185H, but they’re proving incredibly popular with laptop makers: we’re already seeing a flurry of machines built on the chips, and Intel reportedly hopes to sell 100 million Core Ultra PCs by 2025. Note that Intel doesn’t give its NPU an individual TOPS rating, with its 34 TOPS claim based on the CPU, GPU and NPU combined.

QUALCOMM SNAPDRAGON X ELITE



75 TOPS

We had to wait until mid-2024 for the first AI PCs based on Qualcomm’s all-new Snapdragon X Elite to appear, but they promise much. Indeed, at its October 2023 launch, the company claimed it offered “4.5x faster AI NPU processing power than competitors” with its Hexagon NPU capable of 45 TOPS on its own (75 TOPS once you include the CPU and GPU). That figure compared it against the 10 NPU TOPS of the Ryzen 7040, but AMD’s Ryzen AI now offers 16 TOPS – still a big difference.

tom's **HARDWARE**

GET ALL THE ESSENTIAL BREAKING NEWS FOR THE TECH ENTHUSIAST!

No matter if you're building a PC, buying a laptop or learning about robots, Tom's Hardware has all the comprehensive knowledge you need.

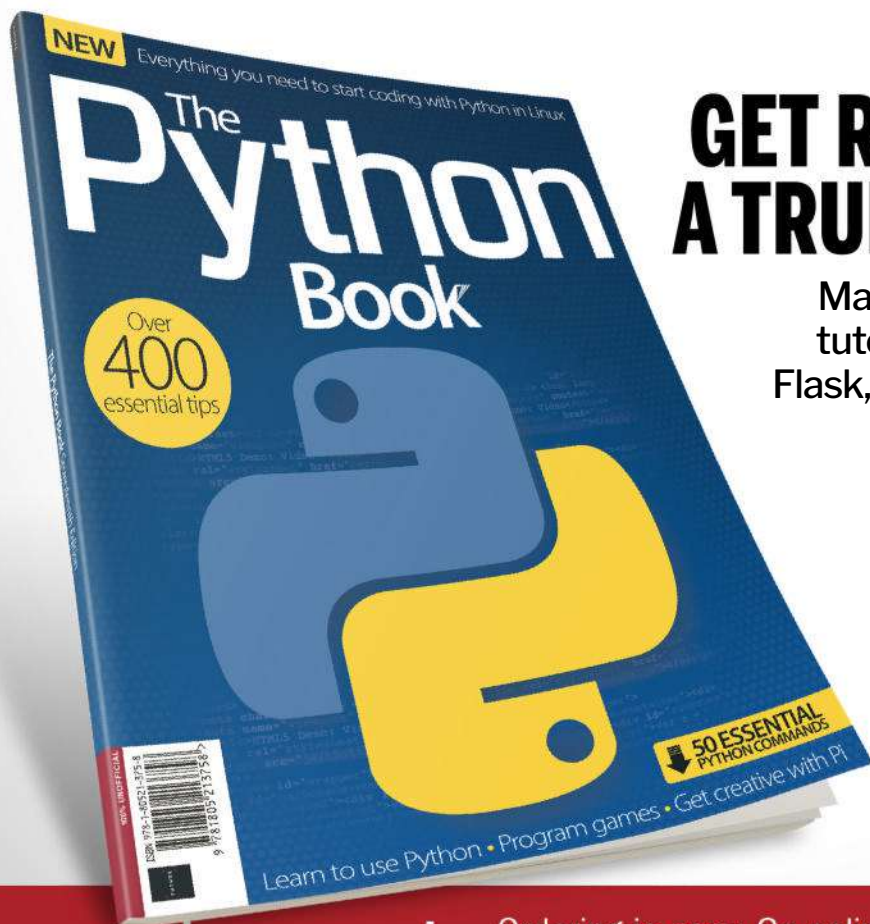


**Scan & Subscribe
for free!**



GET READY TO BECOME A TRUE PYTHON EXPERT

Make Python work for you with tutorials on coding with Django, Flask, Pygame and even more useful third-party frameworks.



**ON SALE
NOW**

Ordering is easy. Go online at:
magazinesdirect.com
Or get it from selected supermarkets & newsagents

HotPicks

VirtualBox » Signal » GpgFrontend » Alpaca »
Keypunch » Falkon » Super Productivity » A-Mazing
Urho » Warzone 2100 » Etcher » Video Downloader



Mayank Sharma

isn't any good with a fishing line, but he knows how to trap and catch the latest and greatest free and open source software.

DESKTOP VIRTUALISER

VirtualBox

Version: 7.0.20

Web: www.virtualbox.org

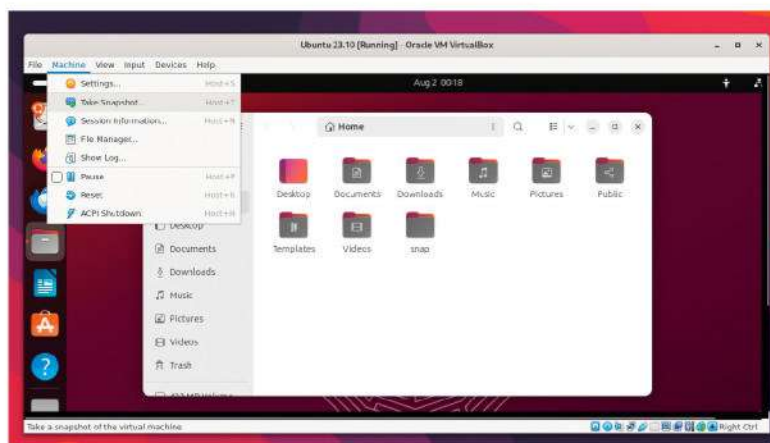
VirtualBox is a wonderful application that has democratised virtualisation and made the quintessential enterprise technology accessible to the average desktop user. Using the app's graphical interface, you can get up and running creating virtual machines in no time.

The VirtualBox project puts out binaries for all popular desktop distros. There's also a RUN script that you can use to anchor the app into your distro. While you're in the Downloads section, remember to grab the universal Extension Pack for the latest release for additional functionality, such as webcam passthrough, disk image encryption and more. To install the extension, head to Tools > Install and point the app to the extensions file.

To create a new virtual machine (VM), head to Machine > New. In the screen that pops up, enter the name of the VM, such as OpenMandriva Rome, and point the app to the distro's ISO image. There are several distros and operating systems, such as Ubuntu, that VirtualBox can install on its own. You do have the option to skip the unattended installation.

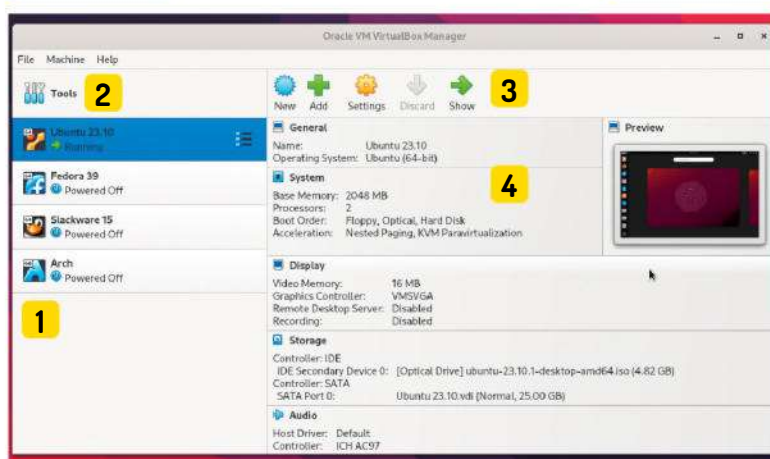
If you go ahead with the unattended installation option, VirtualBox asks you for a few details, such as the login credentials for your new VM, and then installs the distro/OS. If you override the unattended installation, or are installing an unsupported distro, the wizard first asks you for the memory and CPUs you want to assign to the VM. Next you have to create a Virtual Disk for the VM. Hit Finish after you've reviewed the parameters for the VM. If you are new to VMs, it's best to go with the default options.

To boot up the VM, press the Start button. VirtualBox then boots it up just like any physical machine, reads the contents of the ISO images just as a physical computer would read a CD inserted in the tray, and displays the distro's splash screen. From here you can use the virtual machine just as you would use a physical computer.



VirtualBox also enables you to tweak various aspects of the VM while it is running. For instance, you can take a snapshot of the current state of the VM.

LET'S EXPLORE VIRTUALBOX...



1 Virtual machines

This area lists all the virtual machines, along with their current state, such as Running or Powered Off.

2 Tools

Head to this section to create and remove virtual network adaptors. VirtualBox supports host-only adaptors, as well as NAT (network address translation) networks.

3 Create and start VMs

Here you get buttons to create a new VM, and to start the selected VM. In addition to a normal start, you can also start a headless VM in the background.

4 VM details

This area lists all the hardware details about the currently selected VM. If the VM is running, you can also see its mugshot.

SECURE MESSENGER

Signal

Version: 7.16.0

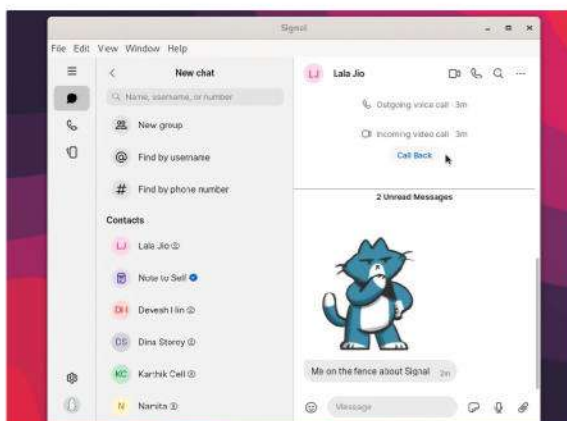
Web: <https://signal.org>

Despite being a centrally controlled and managed messaging app, *Signal* is the apple of privacy enthusiasts' eye due to its strong encryption. You can use *Signal* for one-to-one and group chats, voice and video calls, and file transfers, all of which happen over end-to-end encrypted channels.

Signal works on smartphones running Android and iOS, as well as Mac OS, Windows and Linux. Officially it only supports installation on Debian-based distros such as Ubuntu, Mint and Debian, but it's also available as an unofficial Flatpak, which can be installed with [flatpak install flathub org.signal.Signal](#).

Before you can use *Signal* on your desktop, you have to install the app on your phone. Instead of usernames, *Signal* identifies you with your phone number. After you've installed *Signal* on your phone, unroll the three-dot menu and then go to Settings > Linked Devices.

Now open the *Signal* app on your Linux desktop. It shows a QR code, which you can scan from your phone by tapping the Link A New Device button. *Signal* now



In addition to exchanging text messages, you can also make voice and video calls from *Signal*'s desktop app.

syncs your contacts and groups between your phone and the desktop.

The desktop interface is very similar to the mobile app. You can chat, make audio and video calls, and send photos, videos and audio messages.

To get started, click Compose. The app displays a list of contacts who use *Signal*, in alphabetical order. Just like any other messaging app, you can only engage with someone who is already using *Signal*. You can also tie several contacts into a group by using the New Group option in the Compose page.

Click on a contact to have a one-on-one conversation with them. Inside the chat you also get options to make voice and video calls to them. For added security, you can also enable disappearing messages that self-destruct after they've been read.

ENCRYPTION TOOL

GpgFrontend

Version: 2.1.3

Web: <https://gpgfrontend.bktus.com>

The biggest roadblock to encrypting all your communication is the CLI interface of the tools. *GpgFrontend* solves that problem by hiding the complexities behind a graphical front-end.

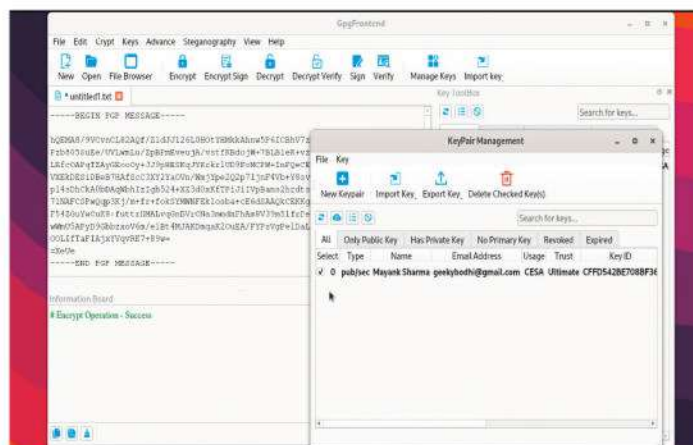
The app is officially distributed on Flathub – install it with [flatpak install flathub com.bktus.gpgfrontend](#).

On first launch, it fires up a brief First Start wizard that tells you how to use the app and points you to the relevant sections of its documentation.

First up, head to Key > Manage Keys to bring up the KeyPair Management window. Then click the New KeyPair button and fill in your details. It's best to just go with the default options for now. We'll use this key to encrypt, sign and verify documents, so it's a good idea to leave all the boxes ticked.

You now need to export your key to the person with whom you want to exchange encrypted documents, and import their key. *GpgFrontend* provides various methods for importing or exporting key pairs.

The best option is to grab keys through a keyserver. In the KeyPair Management window, head to Key >



Import Key > Keyservers. Enter a name or an email ID, and select the server. When you find the key that you're looking for, double-click to import the public key into *GpgFrontend*.

You're now all set to encrypt text and documents. To get started, simply enter some text in the text field. When you're done, select the public key of the user with whom you want to share the text. *GpgFrontend* now encrypts the text. Now send the encrypted text to the person whose public key you used to encrypt the text.

If you receive text encrypted with your public key, copy the encrypted text in the textbox and press the Decrypt button. Now enter the password for your key, and the app decrypts the text.

GpgFrontend makes GnuPG accessible to everyone. But using GnuPG effectively requires some reading, and *GpgFrontend* scores well on this front as well.

DESKTOP AI

Alpaca

Version: 1.0.2

Web: <https://jeffser.com/alpaca>

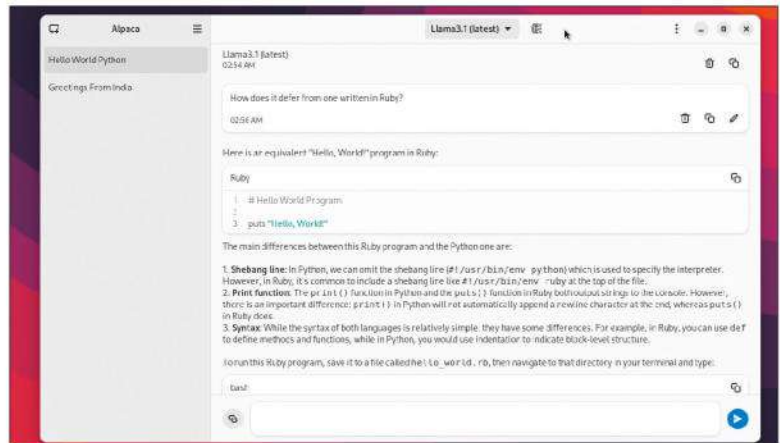
Wouldn't it be cool if you could run an AI model right on your computer? That's exactly what the *Ollama* project is all about. You can use it to download, install and run several open source large language models (LLMs) locally.

However, *Ollama* works on the command-line, and *Alpaca* is the graphical wrapper that does everything you can do with *Ollama*, but using the mouse instead of the keyboard.

The app is officially distributed via Flathub, and you can install it with `flatpak install flathub com.jeffser.Alpaca`. The app greets you with a first-run wizard that shows you a list and asks you to download an AI model into your computer.

The wizard displays four options, and you can choose to download one or all four. Remember that these models weigh several gigabytes and can take a lot of time to download.

But since you can't track the progress of the downloads from the wizard, we advise you to exit it. Instead, click the Manage Models button, which



displays a long list of supported models and their various versions, along with their sizes. Select the one you want to download. This time around, you get a tracker for the downloading model.

Once the model has finished downloading, you can return to the app's main interface. The model is now listed in the drop-down list next to the Manage Models button. If you have downloaded multiple models, you can select the one you want to use from this list.

You can now start interacting with the LLM as you would if they were hosted on the internet. Remember, however, that these models require lots of computing power. Unless you have oodles of RAM and a multi-core machine, getting a response back could take some time.

The models that you download with Alpaca don't have access to the internet, and rely on their training data to come up with responses to your queries.

TYPING TUTOR

Keypunch

Version: 3.0 Web: <https://github.com/bragefuglseth/keypunch>

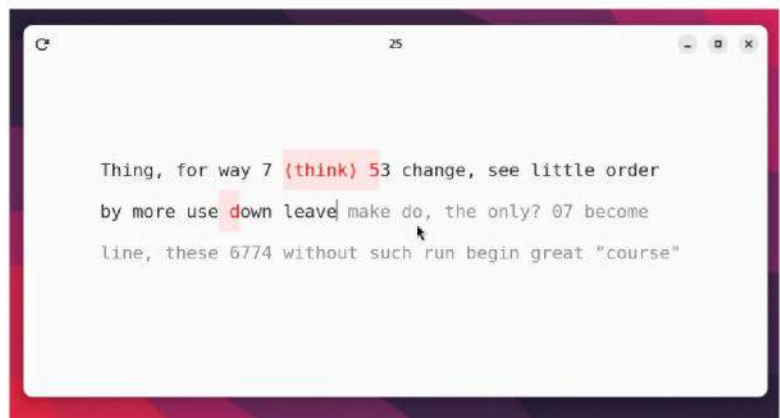
Even if you put on the £3,500 Apple Vision Pro, you'll still not be able to get much work done without typing on a keyword (in this case, a virtual one instead of a physical one). *Keypunch* is a simple typing tutor that can help you increase your speed and accuracy.

The app is officially distributed on Flathub and can be installed with the `flatpak install flathub dev.bragefuglseth.Keypunch` command.

As soon as you launch the app, you're dropped into the middle of a session. The app displays a series of simple, commonly used words, which you have to type without looking at the keyboard. Letters you type accurately are marked in bold, while any mistakes are highlighted in red.

The default exercise is a 30-second Simple session, which begins the moment you start typing. When the time is up, the app displays a result, with your speed in words per minute and your accuracy as a percentage.

Keypunch also displays the session details, such as the session type (Simple), the session duration (30



seconds), and the session language (English). From the main screen, you can use the first pull-down menu to switch to an Advanced session instead, which features numbers, punctuation marks, capitalisation differences and more. You can also alter the session duration from the default 30 seconds down to 15 seconds or all the way up to 10 minutes.

One of the best features of *Keypunch* is that it supports several other languages besides English. You can change the language by navigating to the Text Language option under the hamburger menu. Besides English, the app supports 20 other languages, including Danish, German, Spanish, French, Russian, Hindi and more. Despite this, the application still encourages users to send a request for an unsupported language through its GitHub page.

Keypunch has a simple interface that gets the job done. It doesn't offer any customisation besides the two difficulty levels and support for 20 languages.

WEB BROWSER

Falcon

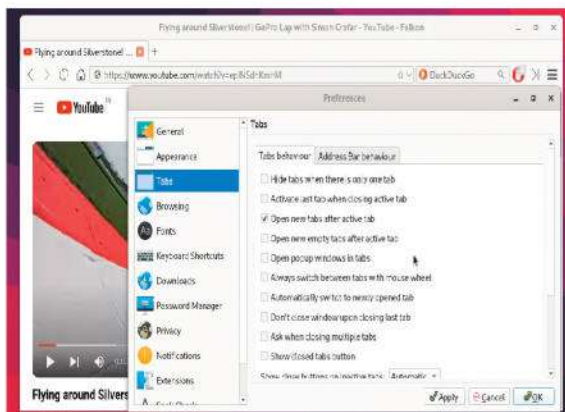
Version: 24.05.2

Web: www.falcon.org

Falcon is a lightweight web browser that's been developed for the KDE desktop. However, because it's officially distributed as a Snap and a Flatpak, you can install it on top of any desktop environment. If you have enabled Snap in your distro, `sudo snap install falkon` installs the browser. If you wish to grab it from Flathub, you can do so with `flatpak install flathub org.kde.falkon`.

As promised, the browser fires up in an instant. It uses DuckDuckGo as its default search engine, and displays *Falkon* and KDE-related websites on its homepage. You can change the URLs of the existing dials and add your own as well.

Falkon doesn't exclude any of the essential web browser features for its lightweight advantage. It can house bookmarks and history, and it includes a spell checker, download manager and password manager, along with an RSS reader, support for multiple tabs, and what have you. The browser doesn't support profiles as such, but there's a session manager that can be used to achieve a similar purpose.



Like the desktop environment it's designed for, *Falkon* has lots of tweakable parameters for several aspects of the browser, such as tabs and address bar.

The browser ships with an ad blocker extension that's enabled by default. *Falkon* ships with about half a dozen more extensions, and you can grab over a dozen more from the internet. This number isn't anywhere close to the number of extensions you get with the mainstream browsers, however.

Still, the extensions that ship with *Falkon* are fairly useful. There's a tab manager, for instance, another extension that displays tabs in a sidebar, another that gives the browser skills to double up as a personal information manager, and more.

You can also customise the appearance of the browser. *Falkon* ships with four themes, and you can grab a few more from the internet. Again, its support for a handful of themes can't rival that of its mainstream cousins, but it does offer some customisability.

TO-DO APP

Super Productivity

Version: 9.0.6

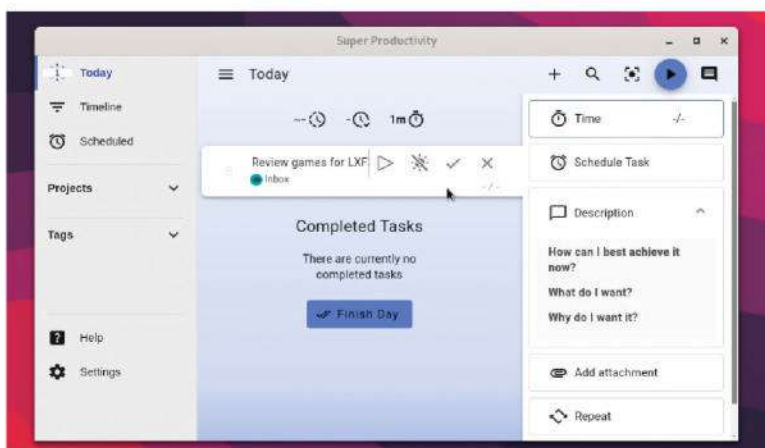
Web: <https://super-productivity.com>

To-do apps come in all shapes and sizes. While you can use *Super Productivity* for keeping track of all kinds of tasks, the app is designed to help developers. It suggests installation through a Snap package, but it also has RPM and DEB binaries, as well as an AppImage. Just grab the AppImage and make it an executable either from the file manager or with the `chmod +x` command.

The app starts with a comprehensive tour around the interface. To begin using it, click the + button in the top to add a task and give it a name. Press Enter to add another task, or Esc to move to the next step.

You can use the additional info button next to a task to view a lot of configurable data. For instance, you can add a description for the task, schedule it for a particular time, adjust time estimates, and a lot more.

When everything's set up, click the Play button next to a task to start tracking the time you spend on it. The



app explains that tracking time can help you optimise your workflow. Press the Pause button to stop tracking time. When you've completed a task, press the tick button to mark the task as complete.

Another standout feature of the app is its ability to import tasks from popular developer platforms, such as GitHub, GitLab and others. The app also has great organisational skills, using which you can group multiple tasks inside a project.

Besides its task management prowess, you can also use the app to help you increase your productivity by employing several techniques such as a Pomodoro timer, a break reminder and others. To configure them, head to Settings and scroll down to the Productivity Helper section.

You can make *Super Productivity* sync to another instance via Dropbox or WebDAV. You can also import and export your data from an instance in a single click.

MAZE

A-Mazing Urho

Version: 20240714 **Web:** <https://luckeyproductions.itch.io/amazingurho>

If you like playing Pac-Man-style maze games, you're going to love *A-Mazing Urho*. You are a fish that spawns inside a maze, which you have to navigate and eat all the shrimp throughout.

Download the zipball from its website and then extract it with `unzip amazingurho-linux-64bit.zip`. Now switch to the extracted folder (`cd amazingurho-linux-64bit`) and launch the game (`./amazingurho`).

You are racing against the clock and to make it even more interesting, you have to feast while avoiding dangers such as hungry squid, crabs and nuclear waste.

Use the left and right keys to turn in that direction and the forward key to swim forwards – remember, you can't swim backwards. This means if you find yourself in a path with a predator at the other end and no forks, you can't swim out of this situation.

The game is still undergoing development, but unlike the first few versions, when you've eaten all the



shrimp in a level, the game then continues to the next level.

The game has two counters on the top of the interface. One displays the number of shrimps you have eaten and the other shows the time left to finish the current level. The game begins with a countdown from three to give you time to prepare. After you've run out of time, you have to press the Enter key to restart the level.

At the bottom of the screen you get a speedometer that indicates your current swimming speed. To get a speed boost, press the movement keys before you swim to the centre of a square. You can use this to either escape a predator or eat shrimp faster.

The *A-Mazing Urho* has some unique predators, including the anglerfish. It roams around the maze with its mouth and swallows you just as you have been swallowing shrimps.

STRATEGY

Warzone 2100

Version: 4.5.1
Web: <https://wz2100.net>

Warzone 2100 was originally released back in 1999 for a limited number of platforms. Once its code was out in the open, development was taken over by a community of developers who ensured the game runs on a wide variety of modern operating systems and on modern hardware.

They've also made several improvements, such as the addition of cross-platform multiplayer options, where you can play with friends on the local network or over the internet.

Warzone 2100 is available as a Flatpak and can be installed with the `flatpak install flathub net.wz2100.wz2100` command.

From the main menu, you start a single-player campaign or a multiplayer one. There's also an option to start local skirmishes with AI bots. If you're new to the game, it's best to start with the tutorial, which will familiarise you with the different HUDs on the screen, such as the command panel.

You can also view the game's intro to understand the backstory. A malfunction in the defence system



caused it to fire nuclear weapons and wipe out most of the civilisation. We are part of an organisation called The Project that's working to make the planet habitable again.

We can play a single-player campaign as part of one of three teams: Alpha, Beta and Gamma. The game has five difficulty levels. One major addition in this version is the inclusion of Fractured Kingdom and Reclamation campaigns.

Our objective is to search for and recover artefacts. We do this by establishing a base by building structures. The game supports several types of structures, which we can build with the help of trucks.

The campaigns involve a lot more than what we can summarise here. But if you are a fan of real-time strategy games, we can assure you *Warzone 2100* will not disappoint.

Warzone 2100 maybe one of the oldest RTS games out there, but its gameplay can rival any of the modern RTS games.

FLASH TOOL

Etcher

Version: 1.19.21

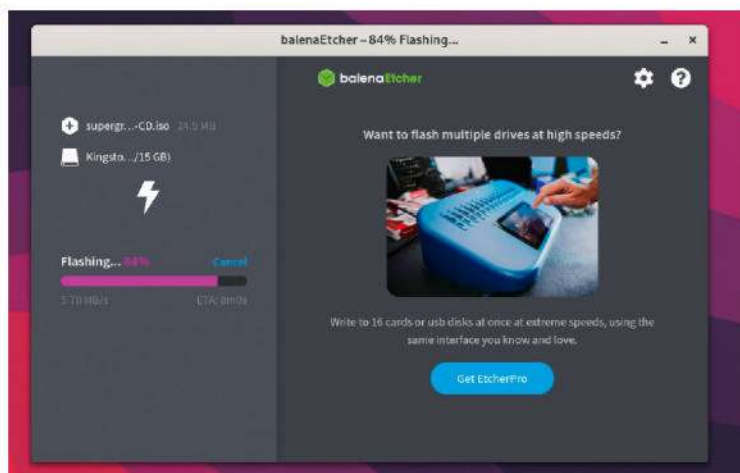
Web: <https://etcher.balena.io>

Most distributions usually ship with a tool to help you write distro images to removable storage media. However, they all have some peculiarities, which has made room for specialised tools such as *Etcher*.

The cross-platform application offers several precompiled binaries for Linux. The recommended option is to download the zipball for Linux and extract it with `unzip balenaEtcher-linux*.zip`. Then you need to change into the folder with the `cd balenaEtcher-linux-x64/` command and, finally, run the app with `./balena-etcher`.

You can also install the app using the RPM and DEB binaries. If you are running Debian or its derivatives, such as Ubuntu, you can install the app with `sudo apt install balena-etcher-*.deb`. If you are running an RPM-based distro, such as Fedora, you can install the app with `sudo yum localinstall balena-etcher-*.rpm`.

The best thing about the app is its simple and appealing interface. The first step is to point it to an image. This can be either an ISO or an IMG file. If you



don't have it on your drive, the app can also fetch it directly from the internet; just point it to the URL.

Instead of writing an image, you can also use *Etcher* to clone an existing bootable disk, because a simple copy and paste wouldn't get the job done.

You have to point the app to the removal storage on which you want it to write the image. Click on Select Target to view a list of all removable media connected to the computer. To ensure you don't accidentally end up wiping your hard drive, the app doesn't show it in the list of targets.

Now hit Flash and the app writes the selected image to the chosen USB stick. Once it's done writing, it also validates the integrity of the drive.

While Etcher is a useful app to flash removable media, it can't carve persistent storage partitions, as you can with your distro's built-in app.

DOWNLOAD VIDEOS

Video Downloader

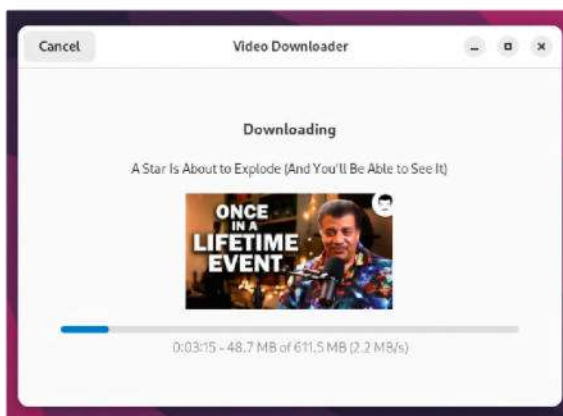
Version: 12.17 Web: <https://github.com/Unrud/video-downloader>

It doesn't make sense to download videos any more. That is until you're sitting inside a metal tube flying at 32,000 feet, or on a boat in the middle of the Pacific. Even if you pay for the expensive internet, the speeds are barely enough to check email.

It's at times such as these you wish you had an offline version of that talk, game or web series you were watching at home, or your favourite music. This is where you need the services of *Video Downloader*.

The app is a graphical wrapper around the popular, but CLI, *yt-dlp* tool, which besides YouTube, can grab videos from virtually any streaming video website. *Video Downloader* is available on Flathub and can be installed with `flatpak install flathub com.github.unrud.VideoDownloader`.

The app has a simple interface. Under the Video tab, enter the URL of the video you want to download. Instead of a single video, you can also point the app to



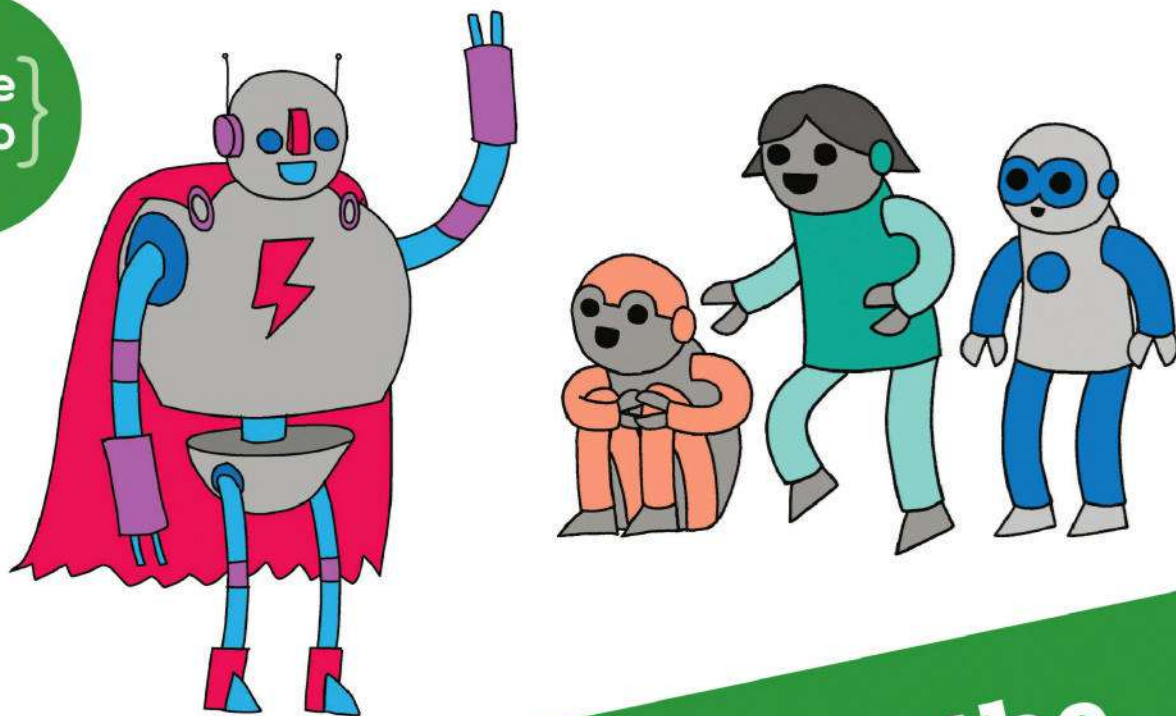
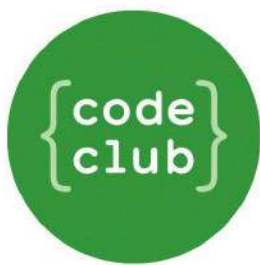
a playlist, and it grabs all the videos within. What you can't do is point the app to multiple URLs at once.

You can also use the Resolution drop-down menu to select the resolution of the videos you want downloaded. The app wouldn't upscale the resolution on low-res videos but can easily downscale videos. For instance, you can download a 4K video as 720p.

Interestingly, the app automatically selects a video format based on the resolution of the video you want to grab, but you can toggle the Prefer MPEG Format option to grab the videos as MP4.

By default, it houses downloaded videos in the `~/Downloads/VideoDownloader` folder. You'll only realise this after the app downloads a video and offers to open the folder it resides inside. **LXF**

The app doesn't offer many controls while downloading a video. You can't pause a download, but you can cancel it.



**Can you help inspire the
next generation of coders?**



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at **www.codeclub.org.uk**

LXF SHELL

Credit: <https://github.com/fritzone/lxf-shell>

Adding colour to the LXF Shell project

Removing his head from the retro demoscene clouds, **Ferenc Deák** is back down to earth and adding colour to our ongoing LXF Shell project.



OUR EXPERT

Ferenc Deák is a colourful chap, bursting with ideas and expertise to not only add a splash of colour to the LXF Shell, but also to make it increasingly useful.

Having taken a well-deserved pause, which we spent in the world of demo colours and pixels, we're ready to honour the pledge we made in the fourth episode of our shell series, and we're back.

Here we stand, poised to delve deeper into the intricacies of the Linux terminal and shell functionality. Not surprisingly after being affected by the colourful series on 90s demoscene effects (see **LXF315-LXF318**), our focus today is on the visually striking features of the Linux terminal, and how we can use them in our shell. So, be prepared for colours and curses.

Beyond black and white

In Linux terminals, printing coloured text is achieved through ANSI escape codes. These provide a means to specify text colours, background colours and text formatting. By incorporating these codes into your commands, you can create visually appealing and informative displays right in your terminal window.

The syntax for using ANSI escape codes is straightforward. Here's a basic example:

```
$ echo -e "\e[31mThis is red text\e[0m"
```

Let's dissect this command:

\e[31m: Sets the text colour to red.

This is red text: Your desired text output.

\e[0m: Resets the text colour to the default.

The output of the command above is the following:

```
> $ echo -e "\e[31mThis is red text\e[0m"
This is red text
```

Not only the foreground of the text can be changed, we can also change the background, too:

```
$ echo -e "\e[31;44mThis is red text on blue background\e[0m"
```

Here's the breakdown:

\e: This is the ANSI escape character, indicating the start of an escape sequence.

[31;44m: This defines the colour attributes – **31** specifies the foreground (red in this case), and **44** is the background (blue in this case). The **m** at the end indicates the end of the colour attribute sequence.

This is red text on blue background: This is the text that will be printed with the specified colours.

\e[0m: This sequence resets the text attributes to the default.

```
> $ echo -e "\e[31;44mThis is red text on blue background\e[0m"
This is red text on blue background
```

There are 16 different foreground and background colours; the very colourful screenshot (opposite page) presents all possible combinations, and the following very short Python program is responsible for this:

```
for foreground_base in (30,90):
    for background_base in (40,100):
        for foreground in range(foreground_base,
                                foreground_base + 8):
            for background in range(background_base,
                                    background_base + 8):
                color_code = f"\033[{foreground};{background}m"
                reset_code = "\033[0m"
                s = str(foreground) + "," + str(background)
                print(f"{color_code}{s}>7}{reset_code}", end=" ")
            print()
```

The C++ version of the same code is not any longer, so we leave that as an exercise you.

Certainly, the Linux terminal is capable of more than just printing out coloured text; it also includes some very basic text formatting functionalities, using the same ANSI sequences. The following list contains the most widely supported formatting capabilities:

\e[1m: Bold or increased intensity.

\e[2m: Faint or decreased intensity.

\e[3m: Italic.

\e[4m: Underlined.

\e[5m: Blink (slow).

\e[6m: Blink (rapid).

\e[7m: Reverse video (swap foreground and background colours).

\e[9m: Strikethrough.

\e[22m: Normal intensity (neither bold nor faint).

And for those who are interested in the subject, the Wikipedia page https://en.wikipedia.org/wiki/ANSI_

QUICK TIP

The code for the shell can still be found at <https://github.com/fritzone/lxf-shell>.

escape_code is the latest and greatest source of information. As an example, let's try to print red, bold, underlined text on white background. Achieving this can be done with the following long sequence:

```
echo -e "\e[1;4;31;47mThis is underlined bold red text on white background\e[0m"
```

The following is the result:

```
1 echo -e "\e[1;4;31;47mThis is underlined bold red text on white background\e[0m"
This is underlined bold red text on white background
```

Now we've had the required introduction on how to write coloured text, it is time to put it to good use.

The almighty prompt

The Linux prompt, often referred to as the command prompt or shell prompt, is a symbol or text string displayed by the shell to indicate that it is ready to accept commands from the user. It typically appears as a cursor or character(s) followed by a space, indicating where you can type commands. The prompt usually contains information about the current environment, such as the username, hostname, current directory or the type of shell being used. Its appearance and content can be customised.

Customising the Linux prompt on the most widely used shells can be done by modifying the PS1 environment variable. We can use special characters and escape sequences to include information such as username, hostname, current directory, time and more. Here are some common escape sequences:

`\u` : Username of the current user.

`\h` : The name of the host the shell runs on.

`\w` : Current working directory.

As our shell already has support for displaying a prompt, and there is even support from one of the plugins (the `dirhandler` plugin we implemented in **LXF313**) to display the current working directory, there is just one task that remains: glue these two together, making the plugins handling the **PLUGIN_PROMPT** class react to a predefined string that will contain the prompt. As our shell does not yet handle environment variables, as a fallback option we will be writing the

30;41	30;42	30;43	30;44	30;45	30;46	30;47
31;40	31;41	31;42	31;43	31;44	31;45	31;47
32;40	32;41	32;42	32;43	32;44	32;45	32;47
33;40	33;41	33;42	33;43	33;44	33;45	33;47
34;40	34;41	34;42	34;43	34;44	34;45	34;47
35;40	35;41	35;42	35;43	35;44	35;45	35;47
36;40	36;41	36;42	36;43	36;44	36;45	36;47
37;40	37;41	37;42	37;43	37;44	37;45	37;46
30;100	30;101	30;102	30;103	30;104	30;105	30;106
31;100	31;101	31;102	31;103	31;104	31;105	31;106
32;100	32;101	32;102	32;103	32;104	32;105	32;106
33;100	33;101	33;102	33;103	33;104	33;105	33;106
34;100	34;101	34;102	34;103	34;104	34;105	34;106
35;100	35;101	35;102	35;103	35;104	35;105	35;106
36;100	36;101	36;102	36;103	36;104	36;105	36;106
37;100	37;101	37;102	37;103	37;104	37;105	37;106
90;40	90;41	90;42	90;43	90;44	90;45	90;47
91;40	91;41	91;42	91;43	91;44	91;45	91;47
92;40	92;41	92;42	92;43	92;44	92;45	92;47
93;40	93;41	93;42	93;43	93;44	93;45	93;47
94;40	94;41	94;42	94;43	94;44	94;45	94;47
95;40	95;41	95;42	95;43	95;44	95;45	95;47
96;40	96;41	96;42	96;43	96;44	96;45	96;47
97;40	97;41	97;42	97;43	97;44	97;45	97;46
90;100	90;101	90;102	90;103	90;104	90;105	90;106
91;100	91;101	91;102	91;103	91;104	91;105	91;106
92;100	92;101	92;102	92;103	92;104	92;105	92;106
93;100	93;101	93;102	93;103	93;104	93;105	93;106
94;100	94;101	94;102	94;103	94;104	94;105	94;106
95;100	95;101	95;102	95;103	95;104	95;105	95;106
96;100	96;101	96;102	96;103	96;104	96;105	96;106
97;100	97;101	97;102	97;103	97;104	97;105	97;106

prompt setup to a configuration file – please read about the widely used JSON format in the boxout (page 93), then we can continue.

For now, to keep things simple, the format of the prompt will be just a simple string, with the following important remarks: to keep some sort of similarity to existing prompts, we will use the escape sequence list from above, with the possibility to add more sequences with different meanings via the plugin interface. Any other textual information will be printed as is.

The complete selection of ANSI colour combinations. Just what everyone needs in their lives!

» 256-COLOURED TEXT

While ANSI escape codes allow for a colour selection consisting of 16 basic colours, the 256-colour mode, supported by several terminal emulators, offers a broader range of hues and shades, allowing for even more vibrant and nuanced displays.

This mode utilises an extended colour palette, with 216 colours defined by RGB values, as well as an additional 16 system colours, plus 24 colours in a greyscale palette. Together, these 256 colours offer unparalleled flexibility for customising your terminal output.

To write 256-coloured text in the Linux terminal, we just need to utilise a specific ANSI sequence, which might look a little different when compared to basic ANSI escape codes. Here's an example:

```
echo -e "\e[38;5;196mThis is a 256-coloured text\e[0m"
```

Let's break down this command to understand it better. Firstly, `\e[38;5;196m` initiates the colour code sequence. The number 196 represents the specific colour from the 256-colour palette.

Next, `This is a 256-coloured text` is your desired text output, while `\e[0m` resets the text colour to the default.

This has the following output:

```
> $ echo -e "\e[38;5;196mThis is a 256-coloured text\e[0m"
This is a 256-coloured text
```

The value 196 is picked from a colour table that can be found, for example, on Wikipedia (https://en.wikipedia.org/wiki/ANSI_escape_code) – the table is, unfortunately (you mean *thankfully?* – Ed), too big to be included. Some terminals also make it possible to output true-colour text; the syntax is similar to the 256-palleted output, except we need to specify the R, G and B values explicitly:

```
echo -e "This is a very \e[1;38;2;96;2;92mbold, Truecolor \e[0m text"
```

Here, **1** indicates the bold selection; **38;2** indicates the true colour output; and **96;2;92** is the RGB value of that lovely purple colour below. The line above produces:

```
> $ echo -e "This is a very \e[1;38;2;96;2;92mbold, Truecolor\e[0m text"
This is a very bold, Truecolor text
```



In order to include colour information in the prompt, we will save you the nightmare of having to manually type in the ANSI sequences for various colours, as done in today's shells, and will resort to a simple list of basic colours: black, red, green, yellow, blue, magenta and cyan, and the default is the one provided by your terminal. For us it is white, as we like white text on a black background, but if you have green text on a grey background, it will be green for you. For this we will introduce two new escape sequences for our shell: `\f{fg_color}` and `\b{bg_color}`. Their usage should be self explanatory: `\fg` specifies the foreground colour, while `\bg` specifies the background colour. To make our life, and the parsing easier, we decided to have the names of the colours in curly braces.

Now we are armed to start and extend the shell to incorporate new functionalities.

Plugging in the prompt

The first new feature is to extend the plugin architecture to handle the recognition of escape sequences from the plugin prompt. We will keep this as generic as possible, so we can reuse the functionality.

The first significant change is the introduction of a new interface element in the plugin architecture:

```
typedef bool PLUGIN_CHECK_ACCEPTANCE_
FPTR(const std::string&);
```

This represents a function pointer to a function that takes in a string and returns a boolean value, true or false. We will use this function in the prompt plugins, to check whether a given prompt handler plugin will accept a given escape sequence.

Introducing this new element also means we need to extend the **PluginDescriptor** class in order to have the corresponding members to store whether the given plugin accepts the given string:

```
struct PluginDescriptor {
...
PluginDescriptor(PluginClass ptype,
const std::string& pfunctionName,
const std::string& pacceptanceChecker):
type(ptype), functionName(pfunctionName),
acceptanceChecker(pacceptanceChecker){}
std::string acceptanceChecker; };
```

We have introduced a new constructor for the class, which also takes a function name representing the name of the function, with the role of accepting or rejecting a specific string token for the current plugin.

This code snippet only shows the difference from the previous implementation. For the full code, please check out the project repositories (*Quick Tip, page 90*).

The next change is in our plugin class, **template <typename C> struct Plugin**, which was extended to provide support for the acceptance function by introducing a new member variable:

```
std::function<PLUGIN_CHECK_ACCEPTANCE_FPTR>
accepter;
```

The next important change happens in the static **PluginContainer createPluginContainer(const std::string& path)** function, which for now resides in **main.cpp**, but we doubt that this will be the case for long, because as our product grows and gets more complex, we will need a much more modular approach.

In the loop, where we are iterating through all the possible plugins loaded, we add a check for whether

we have any value in the **acceptanceChecker** string in the plugin descriptor **pd** (remember the new constructor from a few lines back):

```
if(!pd.acceptanceChecker.empty()) {
promptPlugin->accepter = (PLUGIN_CHECK_
ACCEPTANCE_FPTR*)dlsym(handle, pd.
acceptanceChecker.c_str()); }
```

If there is an acceptance checker using the already presented technique, we load the function name for it and store it in the proper place. The code snippet above presents this for the **promptPlugin** but the logic is the same for all the other plugins.

Now these basic mechanisms are in place, we need to implement the famous acceptance checker function. As an example, we'll continue from last time, when we placed the directory name in the prompt from one of our plugins found in **plugins/dirhandler/cd.cpp**. The first change is the way we create the plugin descriptor for the prompt handling plugin:

```
PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"print_cwd", "check_esc_sequence"}
```

We have specified a third parameter: **check_esc_sequence**. This will be the name of the function implemented in this file (plugin), which will check whether the current escape sequence from the prompt is the one this plugin is handling.

A bit of explanation: as presented above, the prompt is a series of escape sequences and each prompt plugin can handle only one of those. When we display the prompt, we need to iterate through all the prompt plugins, and based on whether they handle a specific escape sequence or not, the text of the prompt is extended with the return value of the plugin function, or for this specific case, the name of the current working directory. We will show the specific code for this later, but for now let's concentrate on the not-so-complex implementation of **check_esc_sequence**:

```
extern "C" bool check_esc_sequence(const std::string&
esc_sequence) {
std::string s = esc_sequence;
s = strim(s);
if(s == "\\w") {
return true; }
return false; }
```

Not so complicated as it seems, the function simply checks whether the parameter **esc_sequence** equals the string **"\w"** or not. If it equals it, it returns true and the piece of code building up the prompt value calls the **print_cwd** in order to include the value of the current working directory in the displayed prompt. If not, it is false and we will not use the plugin function.

The prompt itself

As mentioned in our boxout (*opposite page*), the shell settings will be stored in a JSON file, currently located right where the shell executable is to be found. At this early stage, the following is the layout of this configuration file (called **lxf-shell-config.json**), but this might grow when we expand the functionality:

```
{
"shell": {
"prompt": {
"current":
"\f{green}\u@\h:\f{blue}\w\f{white}$ "
}
```


» JASON, JANICE AND THEIR JSON

At some point, every program needs to store some sort of configuration data. Configuration data can be stored using several file formats, including INI files, JSON, YAML, XML and TOML. INI files offer a simple and human-readable format with sections, keys and values, while YAML offers readability and expressiveness with nested structures. XML supports hierarchical data representation, and TOML prioritises minimalism and readability with key-value pairs and nested tables.

JSON (JavaScript Object Notation) is a lightweight data interchange format commonly used to transmit data between a server and a web application. It's human-readable and easy for both humans and machines to understand. JSON is based on key-value pairs and supports arrays and nested objects, making it easy to represent structured data. Its simplicity and widespread support across programming languages and platforms have made it a popular choice for data exchange in web

development, APIs and many other contexts.

For manipulating JSON data structures, we have decided to use the nlohmann library: <https://github.com/nlohmann/json>.

This is a popular C++ library for working with JSON data. It provides an intuitive and easy-to-use interface for parsing, generating and manipulating JSON data within C++ programs. The library is header-only, meaning it can be included in your project simply by including a single header file,

making it easy to integrate into existing projects.

One of the key features of the nlohmann JSON library is its support for modern C++ features, such as move semantics, initialiser lists and lambda expressions. This allows for concise and expressive code when working with JSON data.

The library provides a rich set of functionalities, such as parsing JSON from strings, streams or files, serialising C++ objects to JSON, and navigating and manipulating JSON objects and arrays.

```
}
}
```

This is not a very complicated structure, it consists of the object shell with a sub-object prompt, having a string value, currently storing the value “\w”.

To include the nlohmann library for reading JSON files, we created an ext subdirectory in our project source tree, then downloaded the library into it:

```
$ git clone https://github.com/nlohmann/json.git
```

We then modified **CMakeLists.txt** to include the new component. The reading of the configuration value for the prompt happens in the following code:

```
std::string retrieveCurrentPrompt() {
    static const std::string default_prompt = "lxfsh$";
    using json = nlohmann::json;
    std::ifstream file("lxf-shell-config.json");
    if (!file) return default_prompt;
    json jsonData;
    try {file >> jsonData;}
    catch (json::parse_error& e){return default_prompt;}
    try { return current_prompt_value =
jsonData["shell"]["prompt"]["current"];}
    catch(json::exception& e){return default_prompt;}
    return default_prompt;}

The incredible flexibility of the library showcased in
return current_prompt_value = jsonData["shell"]
["prompt"]["current"]; retrieves what we need without
too much fuss. The rest is boilerplate for file handling.

The rest of the new code in main.cpp consists of
breaking the prompt into tokens to identify responsible
plugins, but more interesting is building the prompt:
```

```
std::string currentPromptElements =
retrieveCurrentPrompt();
auto promptElements = breakUpPrompt(current
PromptElements);
std::string currentPrompt;
for(const auto& promptElement : promptElements) {
    bool currentElementAccepted = false;
    for(const auto& pc : pluginContainers) {
```

```
for(const auto& p : pc.plugins) {
    if(p->getClass() == PluginClass::PLUGIN_PROMPT) {
        if((dynamic_cast<PromptPlugin&>(*p)).
accepter(promptElement)) {
            std::string promptPluginReturns = (dynamic_
cast<PromptPlugin&>(*p)).handler();
            currentPrompt += promptPluginReturns;
            currentElementAccepted = true; }}
        if(currentElementAccepted) { break; }}
        if(currentElementAccepted) { break; }}
        if(!currentElementAccepted) {
            currentPrompt += promptElement; }}
```

The code iterates through all the prompt elements (the tokens from the prompt) and checks all the prompt type plugins. For the proper prompt plugins, it checks whether they accept the given prompt element.

If the plugin accepts it, we build up the prompt from the return value of the plugin's handler and move on to the next sequence; otherwise we treat the prompt element as plaintext and append to the current prompt.

The colourful plugin

As an example, we provide a plugin to handle all the entries of the colourful plugin, found in **colored_prompt**. The building of the plugin is integrated in **CMakeLists.txt** and contains the following:

```
std::vector<PluginDescriptor> plugins = {
    PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"fg_impl", "check_fg_esc_sequence"},
    PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"bg_impl", "check_bg_esc_sequence"},
    PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"user_impl", "check_user_esc_sequence"},
    PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"host_impl", "check_host_esc_sequence"}
};
```

The implementation of all the functionality can be found in **colored_prompt.cpp**. Next time we'll provide an implementation to provide command history. **LXF**

» **DON'T BE SHELLFISH YOU SHOULD** Subscribe now at <http://bit.ly/LinuxFormat>

Create a VS Code 6502 extension

David Bolton shows how to set up Node, NPM and Yeoman for creating extensions for Visual Studio Code.



OUR EXPERT

David Bolton likes creating 6502 assemblers. Now he's showing how to create an extension in VS Code to run the 6502 assembler he created for Linux Format.

The IDE (integrated development environment) was created in 1983, with Borland's Turbo Pascal considered to be the first example. Before that you edited, compiled and debugged your code with separate programs. It was a clunky process having to switch from the editor to run the compiler and then switch back if a compile error was found. With an IDE, if there is a syntax error, the editor jumps straight to the offending line. With some languages, the errors are identified as you type, without even needing to compile.

Visual Studio Code, also known as VS Code (see the boxout, opposite), has an extension API that can register commands, configurations, key bindings or context menu items, store workspace or global data, display notification messages, use Quick Pick to collect user input, open the system file picker to let users select files or folders, and use the Progress API to indicate long-running operations.

It can also let you change the colours of your source code through theming, change the colours of the VS Code UI or add custom file icons. For now, we're going to produce an extension that can run as65 from a command in the VS Code menu and assemble it.

Extension basics

At its simplest it's just a file, `src/extension.ts`, containing two functions:

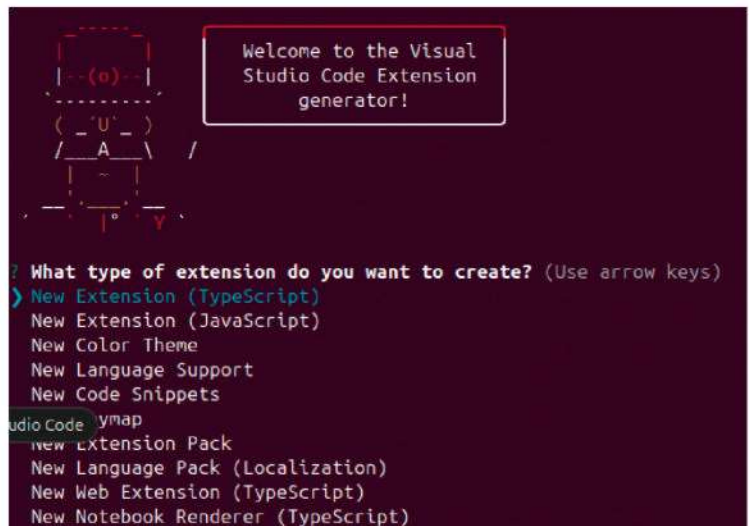
```
export function activate(context: vscode.ExtensionContext): {}
```

And the corresponding but optional:

```
export function deactivate(): {}
```

The idea is that when your extension is first activated, code in the `activate` function registers all the commands specified in `package.json` and provides a mapping for each one in `extension.ts`.

You can write extensions in either JavaScript or TypeScript. Our personal preference is TypeScript,



A successful launch of Yeoman on Ubuntu 24.04.

which if you don't already know is JavaScript with additional syntax added. The TypeScript compiler does static checking for errors before running it, and helps you avoid those silly JavaScript errors at runtime. You can take any valid JavaScript code and put it in a TypeScript file without worrying about exactly how it is written. Simply rename your `.js` files to `.ts`. When you see a file with a `.ts` file extension, that's a TypeScript file.

Getting started

First you need to install VS Code if you don't have it. On Ubuntu, that is done with Snap:

```
$ sudo snap install code --classic
```

For other distros, you need to download an appropriate DEB, RPM or TARGZ and install it with the relevant archiver, see <https://code.visualstudio.com/download>. Now you can run VS Code from a terminal with the command code:

```
$ code
```

Note that after the extension has been created with Yeoman (see later), you can run VS Code with the project name. So, we've used `runas65` for our extension's identifier, which means VS Code opens

QUICK TIP

You can find many extensions with source code on GitHub. Go to <http://github.com> and in the search bar at the top type in `/extension vscode language: TypeScript` or variations of it, like `/vscode-extension language: TypeScript`.

```
? What type of extension do you want to create? New Extension (TypeScript)
? What's the name of your extension? runas65
? What's the identifier of your extension? runas65
? What's the description of your extension? Lets the user assemble 6502 using
the as65 assembler.
? Initialize a git repository? Yes
? Which bundler to use? unbundled
? Which package manager to use? npm
```

Setting the configuration for a new TypeScript extension.

with the following command:

```
$ code runas65
```

Prerequisites for extensions

For extension development, you need both *Node.js* and *Git* installed. *Node.js* is an open source, cross-platform runtime environment for executing JavaScript code. To install it, we first need to install *NPM* (*Node Package Manager*).

If you haven't got *NPM* installed, first you must install *NVM*. This is the *Node Version Manager*. After it's installed, you'll probably have to close and then reopen the terminal.

The command below lets you verify that you have *NPM* installed; it should output `npm`.

```
$ command -v npm
```

Now install *Node* with this command – the `#` means latest version:

```
$ npm install node #
```

Install *Git* if not already installed with:

```
$ sudo apt install git
```

Next, we need *Yeoman* for creating an empty TypeScript or JavaScript project:

```
$ npm install -g yo
```

Followed by the generator:

```
$ npm install -g yo generator-code
```

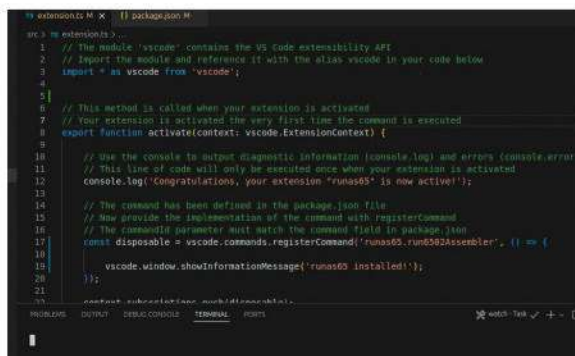
This generates a project with all the common tools installed and installs over 1,000 packages.

Now we can generate the extension with the following command:

```
$ yo code
```

This gives us a choice of various things but only the first two are of interest to us.

We'll go with the default – the TypeScript extension. Press Return, and you're asked a series of questions, starting with the name of the extension. We went with `runas65` – just hit Return to default to that for the identifier, then enter some text as a description. Next, we answered Y to initialise a *Git* repository, then



A freshly minted TypeScript extension, complete with logging.

selected Unbundled for the bundler and selected *NPM* for the packager.

It thought for a bit, added 268 packages, and then said that it found 0 vulnerabilities.

If you press the Return key, it opens VS Code with the folder containing your extension code as the **Workspace** folder. It also offers to install a couple of other extensions for you that might be useful: ESLint and Extension Test Runner, both created by Microsoft.

Check list

The **Workspace** folder should have three folders and a bunch of files inside it. Open the `src` folder in there and you should see `extension.ts`. Click on it to open it in the editor.

Look for the line with `vscode.window`.

`ShowInformationMessage()` and change the message to something like `runas65 installed`. If you used a different identifier, change `runas65` to that.

By default, this starts with a command `runas65.helloworld` installed. Run it by clicking the F5 key. It opens a new VS Code window. Now open the command palette in that window with `Ctrl+Shift+p` and type in `Hello`. You should see it find `Hello World`. Press the Return key. You should now see a popup saying `runas65` is installed.

QUICK TIP

Although the extension adds a command to the VS Code command palette, it is possible to add your own key sequence so you can run the assembler directly from the keyboard while in VS Code. This is covered in the Binding Keyboard Shortcuts To Tasks section on the Tasks page (<https://code.visualstudio.com/docs/editor/tasks>).

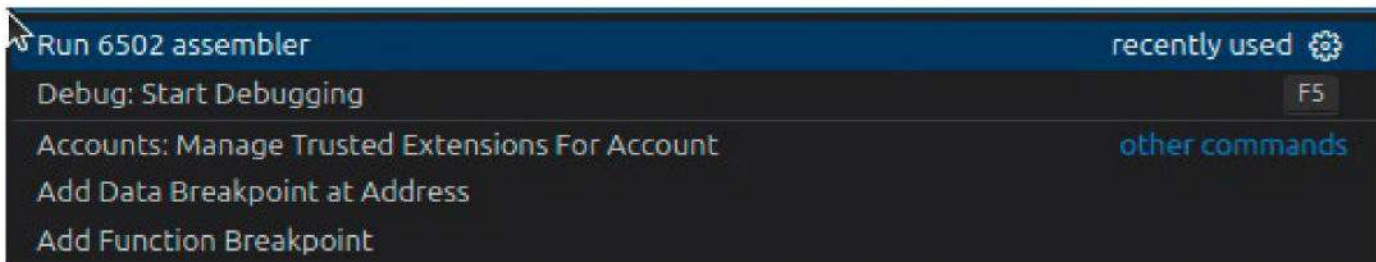
» VISUAL STUDIO CODE

It's one of life's greatest ironies that the company whose former CEO Steve Ballmer branded Linux "a cancer that attaches itself to an intellectual property sense to everything it touches" in 2001 has produced the most popular IDE on Linux: *Visual Studio Code* – according to the PyPI ratings (<https://pypi.github.io/IDE.html>) and others. If you don't like that Microsoft includes telemetry in VS Code (<https://code.visualstudio.com>), you could switch to VS Codium, a community-driven, freely-licensed binary distribution of VS Code.

What makes it so good is it just works nicely and supports – through extensions – every programming language under the sun, or at least those you've heard about. We'd call it an IDE (integrated development environment) but Microsoft more modestly claims it is just a code editor. However, seeing as you can edit, compile and debug code in it (with an appropriate extension), we think it passes the duck test as an IDE: if it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.

Anyone can create extensions for VS Code and in this article we show how to do that and run the 6502 assembler `as65` from LXF319. Download the `as65` zipped source from GitHub <https://github.com/David-H-Bolton/Projects>. Instructions for compiling it are in the `as65` assembler section. This link also has the `runas65.zip` file. VS Code is written in TypeScript and JavaScript, plus HTML and CSS.





Our new VS Code command Run 6502 Assembler.

QUICK TIP

Extensions created by Yeoman also include integration tests automatically. You need to download a command-line tool to run extension tests. There are links to one on the Testing extensions page (<https://bit.ly/lxf320test>).

Close the VS Code window. In the Debug console in the original window, you should see the line `Congratulations, your extension "runas65" is now active`. That line was output by the `console.log` statement on line 11. Feel free to add more `console.log` lines if you want to check if code is run.

You'll see that it's a config file with things like the command title and the command that is run when you select the command and press Return. It also contains the NPM and TSC (TypeScript Compiler) command lines and dependencies.

So let's change the command (line 17) and title (line 18) in `package.json`. The command becomes `runas65.run6502Assembler` and the title becomes "Run 6502 assembler". Line 16 in `extension.txt` becomes `runas65.run6502Assembler`. Now all we have to do is have it run an external command.

Before we do that, let's give it a quick test and press F5 to start it debugging. In the VS Code window that opens press Ctrl+Shift+p and type `run` – you should see Run 6502 Assembler, as in the screenshot (above).

You might see Run 6502 Assembler further down the menu, but after using it a few times, it will move to the top position.

The as65 assembler

Since LXF319, the as65 assembler has had a few more changes. It now has a third option `-l` that creates an assembly language listing so you can see what is produced along with labels, addresses and machine code values. It also has a `test.asm` file and `readme.txt`.

Just a reminder: this is a C# application and you need .NET installed to compile and run it. To install .NET on Ubuntu, just do this:

```
$ sudo apt-get update && sudo apt-get install -y dotnet-sdk-8.0
```

For most other distros, follow the instructions <https://bit.ly/lxf320install>.

Once .NET is installed, do a `cd` in the terminal to where the `as65.sln` and source code are located. This command then builds the assembler:

```
$ dotnet build as65.sln
```

This builds an as65 DLL and executable in `bin/Debug/net8.0` beneath the source code folder. Move the following four files from the `net8.0` folder to `~/runas65/out`:

- `as65`
- `as65.dll`
- `as65.runtimeconfig.json`
- `test.asm`

Run the following command:

```
$ cd ~/runas65/out
```

Then you can test that it works with this command:

```
$ ./as65 test.asm -v -l
```

It should output something like the following into a file `list.txt`, along with a listing of the code and so on.

```
as65 6502 Assembler. Author D. Bolton
```

```
Pass 1
```

```
Pass 2
```

Creating the extension

Now that we've got our skeleton extension created, we have to flesh it out with some code to make it run the 6502 assembler. The starting point for working with external tools is Tasks (<https://code.visualstudio.com/docs/editor/tasks>). Tools can be things like compiler, linters, build systems and terminal-running utilities.

You can run external tools with `ProcessExecution`, shell objects with `ShellExecution` and custom ones with `CustomExecution`. The simplest one is `ProcessExecution` and that's what is used.

This is the code for a Task plus a command to run it:

```
const task = new vscode.Task(  
    {type: 'as65',  
    vscode.TaskScope.Workspace,  
    "as65",  
    "runas65",  
    new vscode.ProcessExecution(exePath,  
    [filename,params],  
    { cwd:as65Path});  
    vscode.tasks.executeTask(task);
```

The `vscode.ProcessExecution` is what runs an external command. We've defined these constants too: `const as65Path = path.resolve(__dirname, '/home/david/as65/bin/Debug/net8.0');`

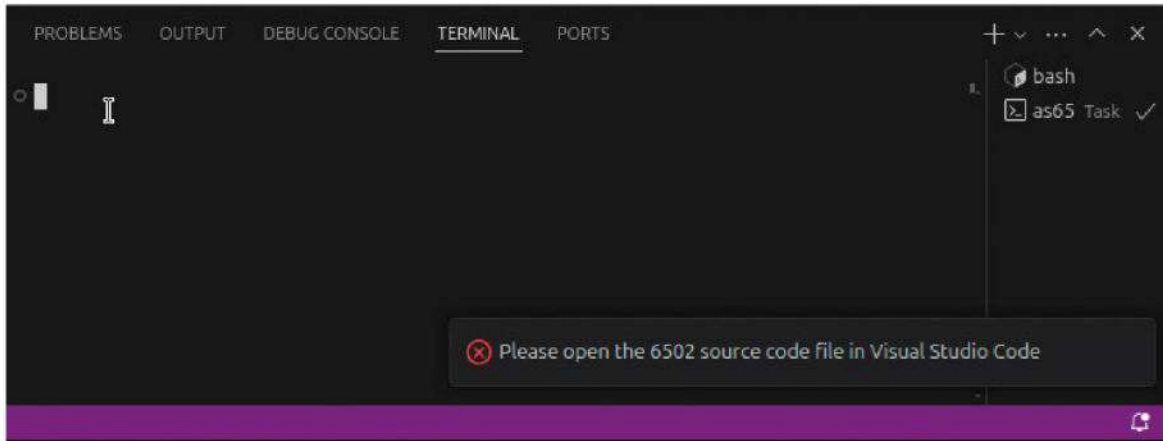
» VISUAL STUDIO CODE EXTENSIONS

VS Code's extensions are what turn it from just a code editor into something much greater. If you want to see what extensions are available, visit the VS Code marketplace (<https://marketplace.visualstudio.com/VSCode>) and click the search button. There are over 61,000 extensions. Click on the All Categories pull-down and you can see how many extensions there are for each category. For programming, there are over 10,200. If you type `6502` in the search box, there are 36 extensions, including several assemblers.

One thing you will find is a degree of repetition. To help you sort the wheat from the chaff, use Sort By and select Installs.

Other CPUs, such as Z80, 6800, 68000 and 8086, also have extensions. Likewise, there are extensions for most programming languages – try C++, Fortran or Pascal in the search box. However, it's the Others category that you might find most interesting, with over 27,700 extensions. Want an Excel viewer or a hex editor, SSH viewer or a utility to convert markdown to PDF? They are all in there.

From the numbers involved, you'll see that it's not difficult to create your own extension and even publish it in the marketplace.



Trying to run the assembler on the VS Code Welcome page.

```
const exePath= path.join(as65Path,"as65");
const params ="-v -l";
```

If you are creating anything more complicated, you should consider putting some of the information in **tasks.json** under the **.vscode** folder. It's hidden by default if you view it in **Files** (all files or folders that start with a **.** are hidden), but VS Code will show it.

That's covered in the tasks link. For simplicity, we've put folders in the code as consts.

The **as65path** says where the utility is located, the **exePath** is the full path to the **as65** and **params** are the params on the end. Add a **-p** to the params if you want it to generate a PRG file.

We also need to pass a filename to **as65** for it to assemble. As we have a code editor, why not assemble the file in the currently focused tab?

```
const filename =vscode.window.activeTextEditor?.
document.fileName;
if (!filename)
{
    vscode.window.showErrorMessage("Please open
the 6502 source code file in Visual Studio Code");
    return;
}
```

The first line gets the currently opened file in VS Code. If you run this on the Welcome page, **filename** is

unassigned and the **if (!filename)** catches it, giving a little popup: 'Please open the 6502 source code file in Visual Studio Code.'

If, however, there is a file open, it passes the path to the Task in **[filename, params]**, which builds the **as65 test.asm -v -l** runs it.

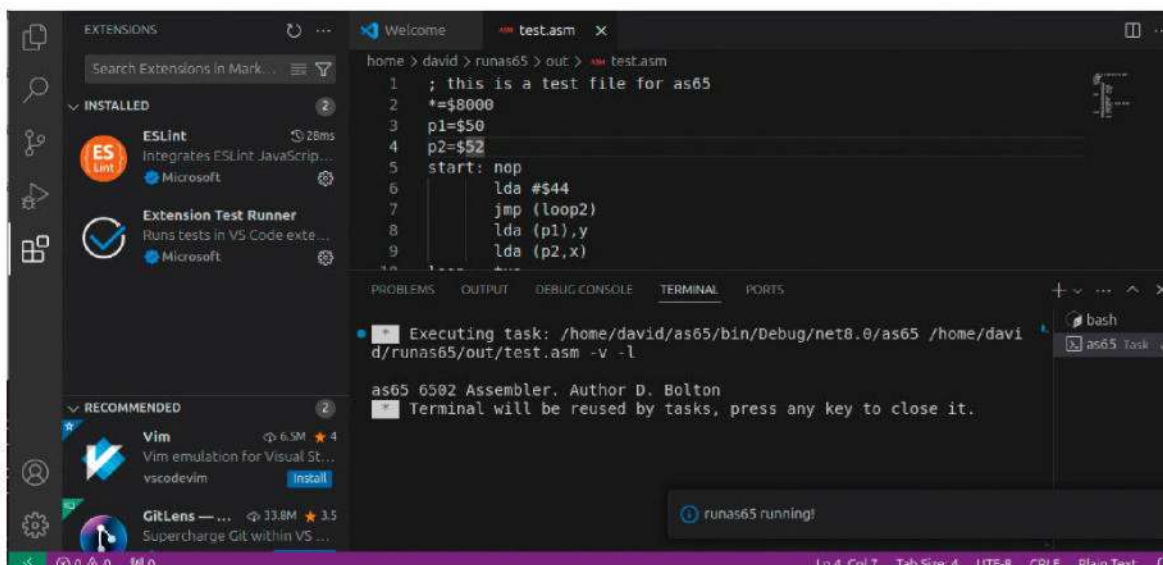
And that is the extension working! Note, on the GitHub page that includes **as65**, you'll also find **runas65**, a complete ZIP of the **extension.ts** and supporting project files.

Now you've seen how to create an extension, go and download the VS Code extensions samples (<https://github.com/microsoft/vscode-extension-samples/tree/main>) – click the Code button on the web page then Download ZIP. This is a 67MB ZIP file with a lot of samples. Then you can create extensions and maybe publish them in Visual Studio Code Marketplace.

If you really want to push the boat out, you can even write debugger extensions that let you debug 6502 code, instead of relying on the *Vice* emulator to run it. That would be quite a bit of work, so is not included here, but it has already been done. Take a look at the **65xx Debugger** (<https://bit.ly/lxf320debug>), a VS Code assembly and C debugger for the 65C02 (a later version of 6502) and 65816 microprocessors. It includes source code on the linked GitHub page. **LXF**

QUICK TIP

VS Code comes with a feature called **Profiles** (<https://bit.ly/lxf320profile>) that lets you customise the editor. Rather than have lots of extensions installed, you can set up different profiles and install, uninstall or disable extensions for each profile, and even change the UI layout. This is stored in the currently active profile.



Successfully running the assembler on the file in the editor window.

» **GET MORE CODING PROJECTS...** Subscribe now at <http://bit.ly/LinuxFormat>

NEXT MONTH

LXF321
will be on sale
Tuesday 15th
October
2024



BUILD A LINUX FORTRESS!

Deploy the ultimate privacy distro: QUBES OS!
Everything virtually secured and fully encrypted.

Listen to the aurora

Find out how cheap hardware and open source can let you tune in and listen to the music of the aurora borealis.

Media everywhere

Discover the best ways to serve your films, music, photos and TV around your home and beyond.

Matter devices

We take a look at the new Google open standard for IoT devices and how you can build your own – hopefully.

Master IRC!

It might be ancient but internet relay chat is still widely used, supported and easier to run than you think.

Content of future issues subject to change. We've locked ourselves out on more than one occasion...

CREDIT: Magictorch

LINUX FORMAT

The #1 open source mag

Future Publishing Limited, Quay House, The Ambury, Bath, BA11 1UA
Email contact@linuxformat.com

EDITORIAL

Editor-in-chief Neil Mohr

Art editor Fraser McDermott

Production editor Katharine Davies

Content director Marc Chacksfield

Group art director Warren Brown

Editorial contributors

Mike Bedford, David Bolton, Neil Bothwick, Stuart Burns, Rob Clymo, Barry Collins, Tim Danton, Ferenc Deák, Nate Drake, Kseniya Fedoruk, Tam Hanna, Matt Holder, Jon Masters, Jody Macgregor, Les Pounder, Michael Reed, Mayank Sharma, Shashank Sharma, Jarred Walton

Cover illustration Magictorch

Ubuntu is a trademark of Canonical Limited.

We are not endorsed by or affiliated with Canonical Limited or the Ubuntu project.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Tux credit: Larry Ewing (lewing@isc.tamu.edu).

Linux Mint is copyrighted 2006 and trademarked through the Linux Mark Institute.

Content production Adequate Media Limited

ADVERTISING

Commercial sales director Clare Dove
clare.dove@futurenet.com

Advertising director Lara Jaggon
lara.jaggon@futurenet.com

Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING

Head of print licensing Rachel Shaw

Linux Format is available for licensing and syndication.

To find our more contact us at licensing@futurenet.com
or view our content at www.futurecontenthub.com

NEW SUBSCRIPTIONS & PAST ISSUES

Web www.magazinesdirect.com

EXISTING SUBSCRIPTIONS

Web www.mymagazine.co.uk

Subscription delays: Please allow up to seven days before contacting us about a late delivery to help@magazinesdirect.com

MANAGE YOUR SUBSCRIPTION ONLINE WITH MYMAGAZINE

Visit www.mymagazine.co.uk/FAQ to view frequently asked questions or log in at www.mymagazine.co.uk

CIRCULATION

Newstrade & retail category director Ben Oakden

PRODUCTION AND DISTRIBUTION

Group head of production Mark Constance

Production manager Nola Cokely

Senior ad production manager Jo Crosby

Digital editions manager Jason Hudson

THE MANAGEMENT

Managing director technology group Paul Newman

Global head of design Rodney Dive

Commercial finance director Tania Brunning

Printed by William Gibbons & Sons

Distributed by Marketforce UK

121-141 Westbourne Terrace, London W2 6JR. www.marketforce.co.uk
For enquiries email: mfcommunications@futurenet.com

Order and access past issues: If you are an active subscriber, you have instant access to past issues through your iOS or Android device/s. Your digital magazine entitlement is available at no additional cost and no further action is required. Pocketmags library may not have access to the full archive of digital past issues. You will only be able to access the digital back issues as long as you are an active subscriber.

To purchase single past issues (print format only): Visit www.magazinesdirect.com (click on Single Issues tab) or email help@magazinesdirect.com. Magazinesdirect.com is owned and operated by Future Publishing Limited.

BAR rates for Linux Format: £84.37 for UK, £171 for Europe, \$194 for USA, £149 for rest of world.

Linux® is the registered trademark of Linus Torvalds in the US and other countries. GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable, code printed in this magazine is licensed under the GNU GPL v2 or later. See www.gnu.org/copyleft/gpl.html. All copyrights and trademarks are recognised and respected.

Disclaimer: All contents © 2024 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future Publishing Limited cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future Publishing Limited and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend and adapt all submissions. All contents in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your systems, peripherals or software through the use of any guide.

We are committed to only using magazine paper derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards.

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

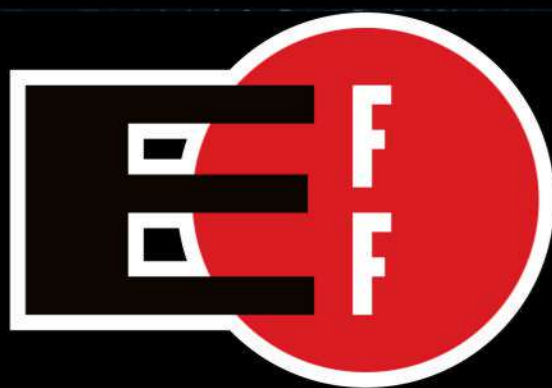


Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR) www.futureplc.com

Chief Executive Officer Jon Steinberg
Non-Executive Chairman Richard Huntingford
Chief Financial Officer Sharjeel Suleman

Tel +44 (0)1225 442 244





The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier



**THE
BRAIN
TUMOUR
CHARITY**

A CURE CAN'T WAIT

BRAIN TUMOURS MOVE FAST. WITH YOUR HELP, WE CAN TOO!

We're working to create a future where brain tumours are curable.
We urgently need your help to accelerate research.

Text DEFEAT5 to 70507 to donate £5, please help us to find a cure.

thebraintumourcharity.org

© The Brain Tumour Charity 2020. Registered Charity in England and Wales
(1150054) and Scotland (SC045081)



Registered with
**FUNDRAISING
REGULATOR**